

## A mobile cloud computing framework for execution of data as a service using cloudlet

Santosh K. Yadav<sup>1</sup>, Rakesh Kumar<sup>2\*</sup>

<sup>1</sup>Dept. of CSE, NITTTR, Chandigarh (U.T), India

<sup>2</sup>Dept. of CSE, CUH, Mahendergarh, Haryana, India

\* Corresponding Author: raakeshdhiman@gmail.com

### Abstract

The mobile device has become an essential utility tool for more effective computation, storage, and power, making it suitable for mobile cloud computing. The cloudlet is used as a connectivity establishment link between the mobile device and the cloud. The objective of this paper is to focus on mobile cloud computing facilitated with cloudlet-based computation. The latter possesses inter-cloudlet communication, which had been proposed within the mobile cloudlet-based computing environment framework. The same had been further enhanced to scalable critical parameter yield of resources framework. Nevertheless, this was not taken to the criteria, which would impact the yield factor, in terms of availability. The present research endeavor aims to improve the algorithm by considering some more criterion and provides a new mobile cloud computing framework for data execution as a service using cloudlet. The outcome shows a positive result in the cloud- cloudlet based computation.

**Keywords:** Cloudlet based computing; data as a service; EAC-SKYR framework; mobile cloud computing; SKYR framework.

### 1. Introduction

The visibly prominent surge in the usage of mobile computing increases the scope of cloud computing in multiple applications. When it is available with complete mobility and without much complexity involved, it is like a cakewalk. Therefore, the concept of cloudlet-based devices which provide cloud services within local access also increases. This indicates the great potential of mobile computing-related fields like mobile cloud computing, cloudlet-based computing, fog computing, 5g technology-based computing, and big data-related technologies. Usage of cloudlets acts in two ways; firstly, it reduces latency while accessing resources from the cloud. Secondly, it improves the computational efficiency of the mobile device. This paper discusses the concerns associated with cloud-cloudlet communication systems and introduces pragmatic solutions to resolve them.

The proposed work utilizes the cloudlet

as a productive service provider, which facilitates the mobile device. The different mobile devices are considered as thin clients because they lack in computing resources, and the cloudlet functions as a resource-intensive mobile device. Thus, a thin client mobile device contacts the cloudlet to enable services within a local premise. If these services are unable to be accomplished by the cloudlet, as mentioned above, then it contacts the cloud via wireless LAN or Wi-Fi to get the task performed for the thin client. The primary objective of accessing information from the cloudlet is its proximity to the client.

### 2. Literature review

(Satyanarayanan, *et al.*, 2009) made a comparison between the working of the cloud and the cloudlets. The former provides services to the client through two levels of hierarchy. The top-level is the cloud, and the

lower level is the thin client. However, the cloudlet provides services in a three-tier hierarchy. This cloudlet is also the 'Clone Cloud' this creates a transient infrastructure of three layers. The clone cloud increases the execution speed by 20 times (Chun, *et al.*, 2011). The cloudlet supports heterogeneity and alleviates the vulnerability of the 'Denial of Service' (DOS) attack (Satyanarayanan, *et al.*, 2014).

Khan, *et al.*, (2015), have declared the cloudlet as a virtual cloud or the Adhoc cloud and used it in their application model for context-aware computing. In edge computing, storage and computing nodes are placed at the fringe of the internet close to sensors and the mobile device. This edge computing promotes 'Internet of Things (IoT) as it provides highly responsive cloud services and assists in the scalability and privacy of various devices (Satyanarayanan, 2017). Another essential aspect is transparent computing, where cloudlet can play a pivotal role. Transparent computing helps multiple client devices to become lightweight with cross-platform capability, enhanced security, and enhanced energy efficiency (Zhang, *et al.*, 2017).

Fog computing is another relevant field-driven from cloud computing technology. Singh, *et al.*, 2019, have discussed the emergence of cloud computing, its merits, drawbacks, and challenges involved. Luong, *et al.*, 2016, elucidated about diverse pricing models which can be applied in cloud computing and in another similar environment that originate from the aforementioned. Pricing models motivate the cloud provider to extend its service to the client.,

As the demand for cloud computing advances, the user wants better management and resource provisioning for high-performance computing. Sri & Narayanan, (2017), have proposed the same along with an automated resource provisioning algorithm to meet the essential requirement of high-performance computing. Khan, *et al.*, (2018), examined analysed varying techniques for evaluating the performance of co-existent heterogeneous wireless networks. They suggested an analytical framework to

evaluate the performance of the same.

This paper analyses numerous frameworks which execute mobile cloud computing using cloudlets. The SKYR framework is also based on cloud-cloudlet communication. The suggested work also refines the performance of the SKYR framework by considering varying parameters that affect the yield factors of availability ( $Y_a$ ). The paper solely focuses on scenario-based execution of data as a service model. The work establishes a relation between the number of mobile devices and the yield factor of availability ( $Y_a$ ). It further increases the overall performance of the SKYR framework for data as a service (DaaS) model and keeps complexity under control.

### 3. Framework for the execution of DaaS

Rawadi, *et al.*, 2014, worked on inter cloudlet communication framework (ICCF) to enable the thin mobile client with cloud services. The ICCF also serves various services like DaaS, network as a service (NaaS), and software as a service (SaaS). However, the framework lacked in centralized management of cloudlets. To overcome this impediment, the ICCF further improved to a framework for mobile cloudlet-based computing (FMCC).

#### 3.1 Framework for mobile cloudlet-based computing

Mobility is the most essential factor to be taken into account in developing a framework for mobile cloudlet-based computing (FMCC) (Artail, *et al.*, 2015). The permits mentioned above communicate among multiple cloudlets, and uses the main server to maintain a directory of a plethora of cloudlets. The main server is known as the root server, is established in the cloud, and is centrally managed. The cloudlets interact among themselves and collectively work on the task delegated by the mobile clients. Generally, the latter approach the root server for a range of services, using mobile networks (3G, 4G, & 5G) and experience greater latency. To subjugate the concern, the cloudlet uses the local wireless LAN to extend

services. One of the major concerns with it is that it acts like a transient device that can go online and offline without any intimation and may result in disruption of services. However, the cloud is at a centralized location and is accessible all the time to get over such occurrences. Mobile clients do not find any cloudlet for functioning; they would be served by the cloud. Hence, it is proven that FMCC solves the difficulty of an inter-cloudlet communication framework with a root server that essentially manages the complete system.

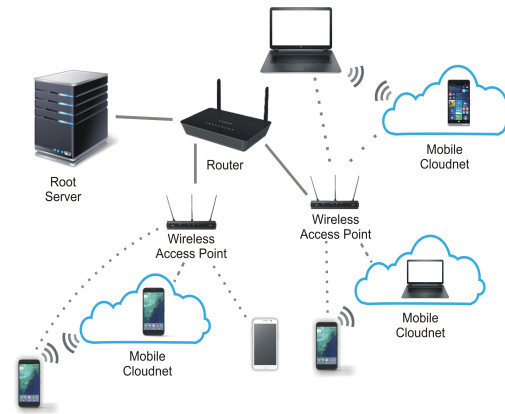
The architecture of cloud-cloudlet communication is as shown in figure 1. Here, the root server is located remotely at the cloud juncture, which handles the interchange between both efficiently. The primary tasks of the root server are to discover new cloudlets, maintain the information about the mobile device along with the cloudlet, and allocate the mobile device to a cloudlet. Following that, the root server establishes direct communication between the mobile client and the cloudlet. The latter enables the former with available resources. If the resource is unavailable, it contacts the cloud to get the same and conserves it for future requests. Hence, serving mobile clients at the local level will save money and time. The FMCC framework can be used for NaaS, DaaS, and SaaS.

Kumar & Yadav, (2017), identified loopholes in FMCC. First of all, the latter lacked a crucial mechanism that was paramount in judging the performance of a system. Following that, it also lacks in the facilitation of management of task accomplishment feedback sent to any mobile device. Thirdly, it was not efficient to provide the completed tasks to the mobile client. Lastly, FMCC is inferior in sensing real-time parameters of the mobile device. Hence, it was not able to consider those while making decisions for potential cloudlets.

### 3.2 SKYR framework

FMCC framework was further modified by Kumar & Yadav in, 2017, where solutions had been proposed for all major issues. The SKYR framework effectively delegates assignments

to multiple cloudlets. It critically senses numerous parameters using a combination of Gabriel architecture (Satyanarayanan, *et al.*, 2014) and large-scale crowdsensing devices (Xiao, *et al.*, 2013). The root server maintains a directory to address the feedback rating concern. A mathematical expression is also added to manipulate several factors. Values of these help in assessing the performance of the cloud-cloudlet system. Figure 2., depicts the



**Fig. 1.** The architecture of cloud-cloudlet communication system.

working model of the SKYR framework. In this architecture, the prime cloud delegates its tasks to small cloudlets, which further serve a huge number of mobile devices. Here, the cloudlets are of two types, the primary and the secondary. The primary cloudlet manages the task assigned to it. Whereas; the secondary cloudlet assists the former in completing those tasks. In figure 2, three primary cloudlets are presented as the primary cloudlet 1, 2 & 3, and nine secondary cloudlets are denoted as S1, S2, S3, S4, S5, S6, S7, S8, and S9. M introduced as master application VM for respective VM cloudlets. The root server establishes this master application VM at the primary cloudlet to manage the task designated to it. This master application VM manages many proxy VMs of respective secondary cloudlets and manages the application VMs running on the secondary cloudlets. The Proxy VM is known by P for sequence-based on secondary cloudlet and application VM is represented as A, B, C, etc. for different applications running on several secondary cloudlets (Kumar & Yadav, 2017).

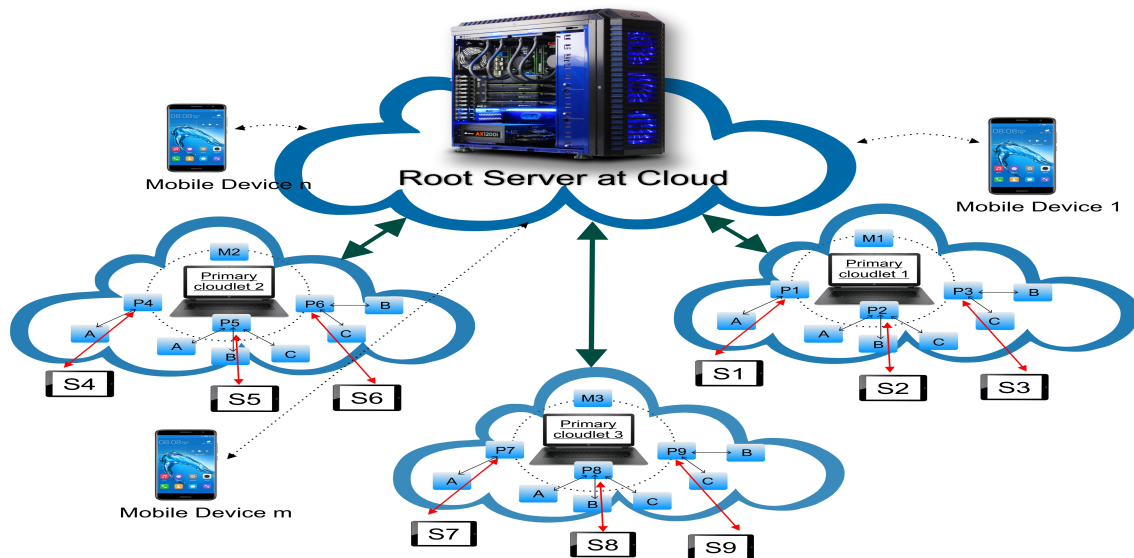


Fig. 2. Working model of SKYR framework

### 3.3 Technical gaps in SKYR framework

Following are the impediments encountered by the SKYR framework and are addressed in the proposed work named as enhanced availability of cloudlet in SKYR framework (EAC-SKYR):

1. The SKYR framework does not emphasize certain aspects, impacting the yield factor of availability ( $Y_a$ ) on these requirements.
2. No algorithm or mathematical expression is given to analyse the dependence of the yield factor of availability ( $Y_a$ ) on the mentioned parameters.
3. SKYR framework also fails to provide the effect of the number of mobile devices on the yield factor of availability ( $Y_a$ ).

### 4. Proposed EAC-SKYR framework

To execute forward the work of the SKYR framework (Kumar & Yadav,, 2017), we have further enhanced the algorithm of evaluating the system's performance by adding some dependent parameters to the yield factor of availability ( $Y_a$ ).

$$\text{Yield factor for availability } (Y_a) = \left\{ 1 - \left( \left( \frac{x}{y} \right) \times t \right) \right\} \quad (1)$$

In equation 1, 'x' represents the total number

of instances when a particular cloudlet is not available for computing, and 'y' represents the total number of instances a remote root server is searched for various cloudlets during a particular time duration denoted as 't..' Equation 1 depicts that when the value of  $Y_a$  is '0', it means a particular cloudlet is not available even for a single instance within a specific time duration 't..' The value of this time duration can be selected from table 1.

Similarly, when  $Y_a$  is '1' during a specific time duration 't' it suggests that a particular cloudlet is available in all searched instances. Equation 1 helps predict the performance of a cloudlet based on the yield factor of availability ( $Y_a$ ). Hence, the  $Y_a$  factor is significant. When it is '0' or nearly '0' during a specific duration 't' for a particular cloudlet, then that cloudlet's permission to impart services can be revoked. Similarly, if  $Y_a$  factor is '1' or nearly '1' for a particular cloudlet during a specific time duration 't' then that particular cloudlet will enjoy its permission to impart services to thin clients. According to the SKYR framework, a long-time duration should be considered for better results. So, the value of 't' is regarded as 1 in all practical scenarios.

It is important to calculate the  $Y_a$  for every cloudlet so that the root server can remove the inferior quality cloudlets. Parameters that influence the yield factor of availability ( $Y_a$ )

**Table 1.** Probability values for different time durations (Kumar & Yadav, 2017).

S. no.	Time duration	Probability value for the time duration (t)
1.	Very long duration	1
2.	Long duration	0.8
3.	Medium-term	0.6
4.	Short term	0.4
5.	Very short term	0.2
6.	Immediate	0

**Table 2.** Presumed values for various factors on which yield factor of availability ( $Y_a$ ) is dependent.

S. no.	Parameter	Denoted as	Presumed value	Threshold potential for the respective parameter
1.	Power	p	1	> 30% Battery back up
			0	< 30% Battery back up
2.	Signal strength	q	1	> 20% Signal intensity
			0	< 20% Signal intensity
3.	Spatial movement	r	1	Less frequent movement (stable)
			0	Very frequent movement (unstable)
4.	Processor availability	s	1	> 50% CPU utilization availability for external tasks
			0	< 50% CPU utilization availability for external tasks

are power (denoted by  $p$ ), signal strength ( $q$ ), spatial movement ( $r$ ), and processor availability ( $s$ ). These four parameters have a significant impact on the yield factor of availability for any cloudlet.

**Power ( $p$ ):** It measures the availability of battery backup or online power backup of cloudlet.

**Signal strength ( $q$ ):** It suggests the intensity of the signal of the cloudlet through which communication will take place. These signals can be of Wi-Fi, 3G, 4G, and 5G, etc.

**Spatial movement ( $r$ ):** This factor depicts the mobility of cloudlets or how frequently it changes the location.

**Processor availability ( $s$ ):** It indicates the availability of cloudlets. If a device has a smaller number of processes to execute, it can execute the tasks from other mobile clients.

The proposed improved SKYR framework is named as enhanced availability of cloudlet in SKYR framework (EAC-SKYR), as it improves the availability of cloudlet to a great extent as compared with the SKYR

framework. Monitoring the above-said parameters for cloudlets significantly improves the availability of cloudlets and hence improves the computational efficiency of cloudlet-based computing systems.

These parameters suggest that when all four parameters are '1' only then cloudlet can impart its services as a cloudlet. In other words, we can say that when all the four parameters are '1', then the value of 'x' will be '0', and in the rest of all scenarios, its value will be '1'. This condition can be represented in the equation as

$$\mathbf{x} = \overline{p} + \overline{q} + \overline{r} + \overline{s} \quad (2)$$

The same equation can also be described as,

$$\mathbf{x} = \overline{pqrs} \quad (3)$$

This suggests that the value of 'x' will be '1' in all scenarios except when all parameters  $p$ ,  $q$ ,  $r$ , and  $s$  hold the value '1'. Value of 1 for 'p' means cloudlet has sufficient battery power to assist. Similarly, when 'q' is '1' means signal strength is significant to operate. Value of '1' for 'r' means spatial movement is not much



fast, and the device is not changing its location very promptly. Similarly, when ‘s’ is ‘1’, it suggests that the processor is available for computing services.

From table 2, we can get the values for various parameters  $p$ ,  $q$ ,  $r$ , and  $s$  on which yield factor of availability ( $Y_a$ ) depends. These values are presumed while executing scenarios in Cloudsim. These potential threshold values for various parameters can be changed as per the requirements in a practical environment. Figure 3 shows the flow chart for a cloudlet. If there are ‘n’ cloudlets in the system, the same logic can be repeated to check the performances of all cloudlets.

Pseudocode to control the performance of the cloudlets

```

If ( $0.8 \leq Y_u \leq 1$ ) or ( $0.8 \leq Y_o \leq 1$ )
{
    Cloud-cloudlet computing system is working fine
}
Else
{
    If ( $Y_u < 0.8$ )
    {
        The problem of under-provisioning, the root server will discover new cloudlets
    }
    Else
    {
        A case of over-provisioning.
        For ( $k=1$  to  $n$ ,  $k++$ )
        {
            If ( $Y_{ak} \geq 0.8$ )
            {
                Retain cloudlet
            }
            Else
            {
                If ( $Y_{ak}(p)=0$ ) or ( $Y_{ak}(s)=0$ )
                {
                    Send a warning message to cloudlet  $k$ 
                }
                Else
                {
                    Discard cloudlet  $k$ 
                }
            }
        }
    }
}
}
}
}

```

In the given pseudo-code, ‘n’ denotes the number of cloudlets whose performance need to be checked, ‘k’ is the counter variable which will trace all cloudlet from 1 to n. So, for all cloudlets, the yield factor of availability ( $Y_a$ ) is checked, and if it lies in range then, it will be retained. Otherwise, factors like power ( $p$ ) and processor availability ( $s$ ) will be checked for that cloudlet, ; if it is ‘0’, it means cloudlet is not available due to power or processor availability problem. So, in this case, a warning message is sent to the cloudlet to improve its availability. If both values are ‘1’, that means cloudlet is having some other problem. In that case, it will be discarded from the pool of cloudlets. In the pseudocode,  $Y_{ak}$  denotes the yield factor of availability for kth cloudlet. Similarly,  $Y_{ak}(p)$  and  $Y_{ak}(s)$  denote the yield factor of availability in the context of power and processor availability parameters for the kth cloudlet.

### 5. Performance comparison of individual cloudlets

The proposed work is implemented in Cloudsim as existing works of SKYR framework (Kumar & Yadav, 2017) and FMCC framework (Artail, *et al.*, 2015) were also executed in the same environment. So, the same scenarios and assumptions are followed in the proposed work. Representative parameters of cloudlets are shown in table 3.

**Table 3.** Representative parameters for cloudlets

Parameter	Value
MIPS	10,000 MIPS
RAM	2GB
Frequency when Idle	300MHz
The frequency at high CPU	1.9GHz
CPU cores	2
Storage size	16GB
Battery capacity	2000mAh
Watt-hours capacity	8Wh

Here, three practical scenarios are considered and are shown in table 4. However, in all three cases, we have selected cloudlet 1 and cloudlet 2 as a sample to be monitored among a set of

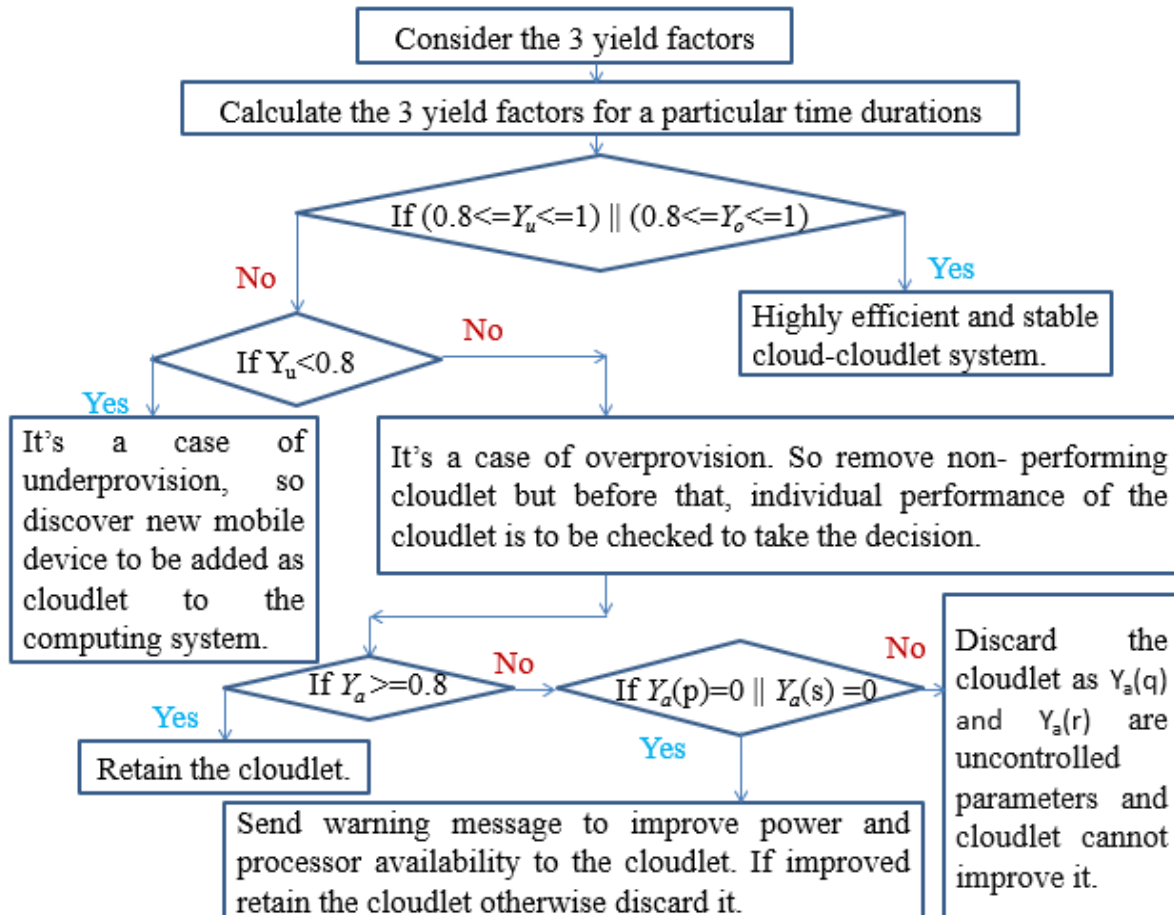
cloudlets. The result shows the average of all 3 executed cases.

**Table 4.** Practical scenario for EAC-SKYR framework

Cases	Number of cloudlets	Number of mobile devices	Time duration (minutes)
Case 1	5	20	50
Case 2	10	40	
Case 3	15	60	

In (Kumar & Yadav, 2017), a comparison of two individual cloudlets is done by comparing the yield factor of availability for the cloudlets denoted as  $Y_{a1}$  and  $Y_{a2}$ . As we have discussed that the yield factor of availability ( $Y_a$ ) is dependent on factors like power ( $p$ ), signal strength ( $q$ ), spatial movement ( $r$ ), and processor availability ( $s$ ).

When we consider these factors in the performance evaluation of two individual cloudlets keeping all scenarios and assumptions the same, which will result in the graph shown in Figure 4.s From this graphical comparison, we can depict that  $Y_{a1}$  (yield factor for the availability of cloudlet 1) is a combined curve of all four factors  $Y_{a1}(p)$ ,  $Y_{a1}(q)$ ,  $Y_{a1}(r)$ , and  $Y_{a1}(s)$  curves, which represent power, signal strength, spatial movement, and processor availability respectively for cloudlet 1. Similarly, for the 2nd cloudlet, the individual curve is plotted for  $Y_{a2}(p)$ ,  $Y_{a2}(q)$ ,  $Y_{a2}(r)$ , and  $Y_{a2}(s)$  and which in result gives the combined curve for  $Y_{a2}$ .  $Y_{a1}$  and  $Y_{a2}$  curves are denoted with a solid line, and other factors on which yield factor of availability ( $Y_a$ ) depend are represented with dotted lines. So, this graph shows that the yield factor of availability is



**Fig. 3.** The logical flow of the EAC-SKYR framework using various yield factors and other parameters

heavily dependent on these four factors. Table 5 shows the comparison of features of various cloudlet-based computing frameworks.

### 6. Comparison of EAC-SKYR framework with existing SKYR and FMCC framework

In this section, a comparison of the proposed EAC-SKYR framework with the SKYR and FMCC framework, in terms of power consumption and processor accessibility is made. The research also improves the existing framework by monitoring the regularity and

performance efficiency of the cloudlets by sending warning messages accordingly.

The signal strength and spatial movement are the contributors, that cannot be controlled directly, but closely scrutinizing these certainly aids in analyzing the conduct of cloudlets. It also permits the EAC-SKYR framework to have better control over cloudlet-based computing. The results depict that the proposed framework performs more effectively, about CPU utilization, processing delay, energy usage, overall delay, and average burden on the server.

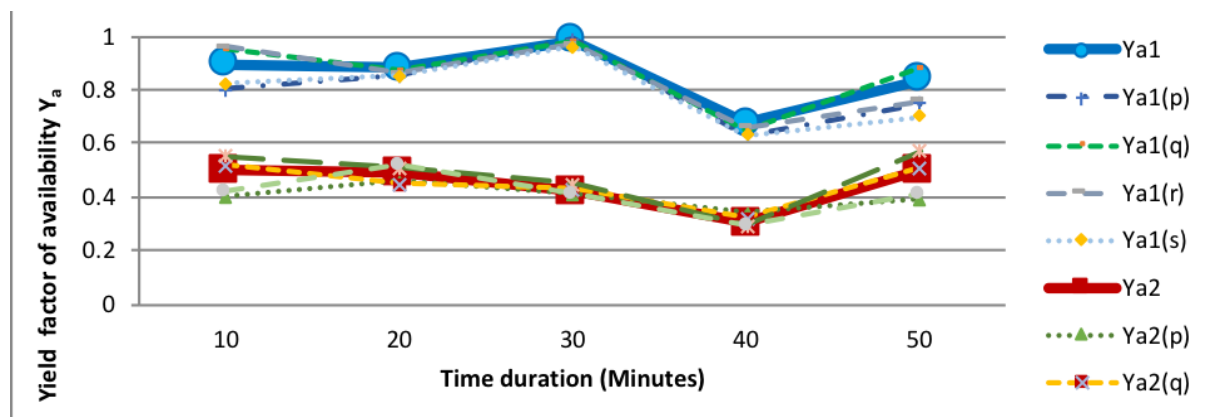


Fig. 4. Effect of a various factor on the performance of two individual cloudlet's yield factor of availability Ya1 and Ya2.

Table 5. Features of a various framework of cloudlet-based computing .

S. no.	Parameters	ICCF	FMCC	SKYR	EAC-SKYR
1.	Centralized control	No	Yes	Yes	Yes
2.	Intercloudlet based communication	Yes	Yes	Yes	Yes
3.	Support for DaaS	Yes	Yes	Yes	Yes
4.	Support for NaaS	No	Yes	Yes	Yes
5.	Support for SaaS	No	No	Yes	Yes
6.	Scalability support for cloudlet	No	No	Yes	Yes
7.	Performance evaluation feature for cloudlet	No	No	Yes	Yes
8.	Delegation of the task to multiple cloudlets	No	No	Yes	Yes
9.	Consideration of real-time parameter for performance evaluation	No	No	No	Yes
10.	Scalability with real-time parameter	No	No	No	Yes



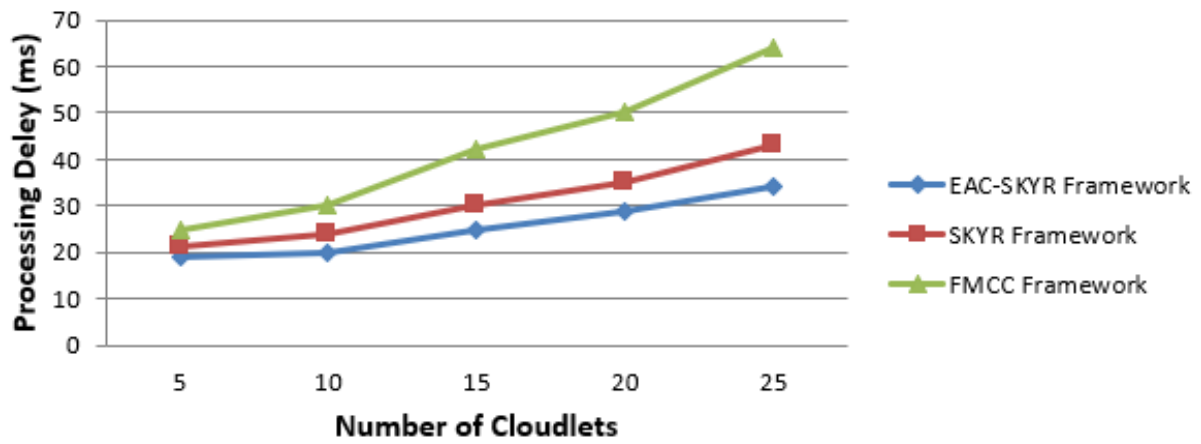


Fig. 5. Processing delays versus the number of cloudlets

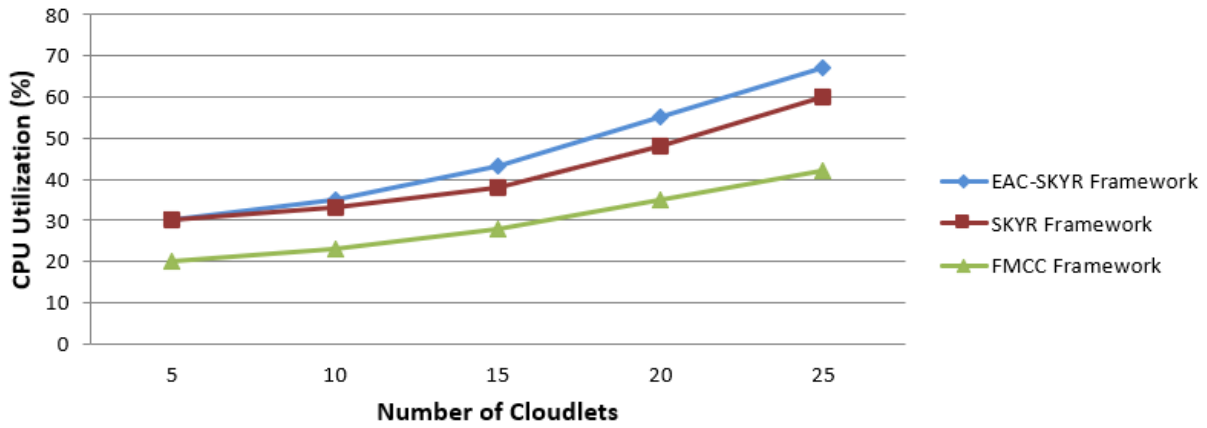


Fig. 6. CPU utilization versus the number of cloudlets.

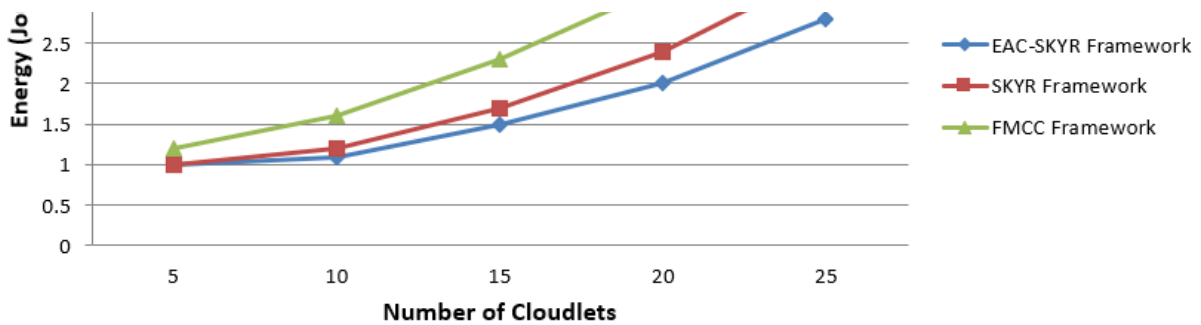


Fig. 7. Energy consumption versus the number of cloudlets

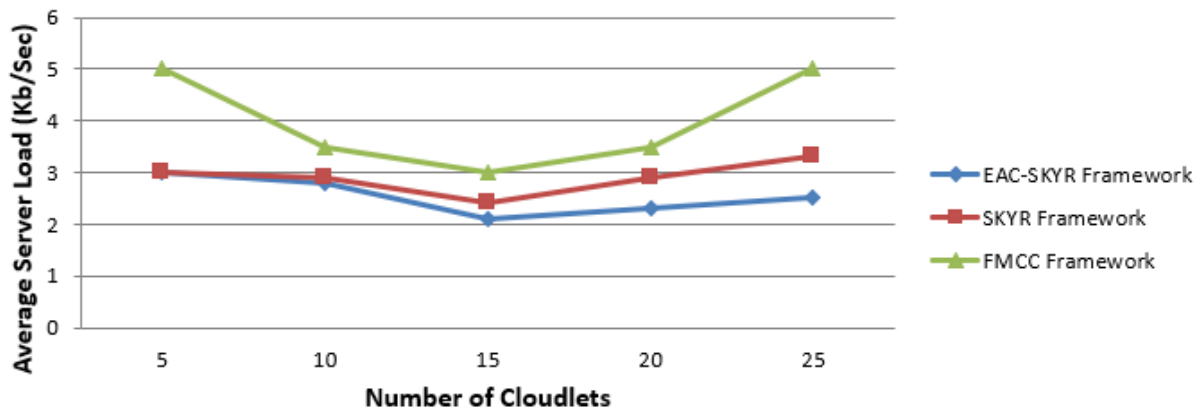


Fig. 8. Server loads versus the number of cloudlets

Table 6. Summarized result comparison of various frameworks.

Parameters	Various cloudlet-based computing framework											
	FMCC				SKYR				EAC-SKYR			
	Number of cloudlets used											
Processing delay (ms)	10	15	20	25	10	15	20	25	10	15	20	25
CPU utilization (%)	30	42	50	64	24	30	35	43	20	25	29	34
Energy usage (Joules)	23	28	35	42	33	38	48	60	35	43	55	65
Server load (Kb/Sec)	1.6	2.3	3.2	4.5	1.2	1.7	2.4	3.4	1.1	1.5	2	2.8
	3.5	3	3.5	5	2.9	2.4	2.9	3	2.8	2.1	2.5	2.6

### 6.1 Processing delays

As the SKYR framework is scalable and maintains that order, the suggested EAC-SKYR framework allows cloudlets to join and quit the cloudlet-based computing system by their choice. As the number of cloudlets rises in the system, it helps to facilitate more result-oriented service to the mobile device. However; it also advances the load on the framework to manage the number of cloudlets by keeping a fixed number of mobile devices. The performance of the proposed EAC-SKYR framework is compared to the existing framework as shown in figure 5.

### 6.2 CPU utilization

The necessity of cloudlets surges with the rise in several mobile devices. Following measurement of the CPU utilization by a varied number of mobile devices. As the number of cloudlets gets enhanced, there is an elevation in CPU usage. Thus, the efficient management of the framework assists in enhancing the utilization of the above stated, as is further shown in Figure 6

### 6.3 Energy consumption

The CPU functioning plummets with the

multiplying of cloudlets, which also causes a hike in energy consumption. Even after increasing the number of cloudlets, energy usage in the suggested EAC-SKYR framework is still more effective than the existing framework, on account of impactful CPU exercise. Energy usage is directly proportional to the CPU-related operation, as elucidated above and illustrated in figure 7.

#### 6.4 Average server load

With the addition of a greater number of cloudlets, more pressure would be exerted upon the server. The overall delay is reduced in the proposed framework that raises the availability for computing. This further results in the alleviation of temporary cloudlet-based information. Thus, enhancing the performance of the server. In the nascent stage, the server load is more, but with the increase in the number of cloudlets, it decreases, as is visibly evident in figure 8. After achieving the optimum level, the server's load starts surging again, as it is supposed to maintain the information of numerous cloudlets. The complete summary of the comparison between different frameworks is depicted in table 6.

### 7. Conclusion

Overall, it can be appropriately summed up that a mobile cloud computing framework for DaaS is proposed that is an extension of the SKYR framework (Kumar & Yadav, 2017). In this paper, varied criteria are considered, affecting the yield factor of availability (Ya), and an improved algorithm is proposed. The work is executed in the Cloudsim simulator. The results depict that the proposed EAC-SKYR framework performs better than the existing frameworks.

In the future, an effort will be made to focus on the sensing aspect of a varied real-time criterion of cloudlets that may further improve the performance of the SKYR framework. An intrinsic pricing model may also be advocated in the future to motivate a diverse range of mobile devices to act as a cloudlet.

### References

- Artail, A., Frenn, K., ; Artail H. & Safa, H. (2015).** A framework of mobile cloudlet center based on the use of mobile devices as cloudlets. Proceedings of 29<sup>th</sup> IEEE international conference on advanced information networking and applications. Gwangju, South Korea, 777-784.
- Chun, B., ; Ihm, S., ; Maniatis, P., ; Naik, M. & Patti, A. (2011).** CloneCloud: Elastic execution between mobile device and cloud. Proceedings of 6<sup>th</sup> conference on computer systems (EuroSys). Salzburg, Austria, 301-314.
- Khan, A., ; Othman, M., ; Xia, F. & Khan, A.N. (2015).** Context-aware of mobile cloud computing and its challenges. IEEE Cloud Computing, 2(3): 42-49.
- Khan, B., ; Ahn, J., ; Park, E., (2018).** An analysis framework for the performance of collocated heterogeneous wireless networks with negative acknowledgments. Kuwait Journal of Science, 45 (1): 70-78.
- Kumar, R. & Yadav, S.K. (2017).** Scalable Key Parameter Yield of Resources Model for Performance Enhancement in Mobile Cloud Computing. Springer's Wireless Personal Communications, 95(4): 3969-4000.
- Luong, N.C., ; Wang, P., ; Niyato, D., ; Yonggang, W. & Han, Z. (2016).** Resource Management in Cloud Networking Using Economic Analysis and Pricing Models: A Survey. IEEE Communications Surveys & Tutorials, 19(2): 954-1001.
- Rawadi, J. M., ; Artail, H. & Safa, H. (2014).** Providing local cloud service to mobile devices with intercloudlet communication. Proceedings of 17<sup>th</sup> IEEE Mediterranean electrotechnical conference. Beirut, Lebanon, 134-138.

**Satyanarayanan, M., ; Bahl, P., ; Caceres, R. & Davies, N. (2009).** The case for VM-based cloudlets in mobile computing. *IEEE Pervasive Computing*, **8**(4): 14–23.

**Satyanarayanan, M., ; Chen, Z., ; Ha, K., ; Hu, W. & Richer W., et al. (2014).** Cloudlet: At the leading edge of mobile-cloud convergence. Proceedings of 6<sup>th</sup> international conference on mobile computing, application, and services (MobiCASE). Austin, Texas, USA, 1-9.

**Satyanarayanan, M. (2017).** The emergence of edge computing. *IEEE Computer*, **50**(1): 30-39.

**Singh, S.P., ; Nayyar, A., ; Kumar, R., & Sharma, A. (2019).** Fog computing: from architecture to edge computing and big data processing. *The Journal of Supercomputing*, **75**(4): 2070-2105.

**Sri, L., ; Narayanan, B., (2017).** Speculation resource provisioning in high-performance computing. *Kuwait Journal of Science*, **44** (1): 58-63.

**Xiao, Y., ; Simoens, P., ; Pillai, P., ; Ha, K., & Satyanarayanan, M. (2013).** Lowering the barriers to large-scale mobile crowdsensing. Proceedings of ACM conference. Jekyll Island, Georgia, 1-6.

**Zhang, Y., ; Guo, K., ; Ren, J., ; Zhou, Y., ; Wang, J. & et al. (2017).** Transparent computing: A promising network computing paradigm. *IEEE Computing in Science & Engineering*, **19**(1): 7-20.

**Submitted:** 20/04/2020  
**Revised:** 10/08/2020  
**Accepted:** 09/11/2020  
**DOI:** 10.48129/kjs.v48i3.9584