# Security And Privacy Optimization And Service Provider Selection For Cloud Computing For Small And Medium Educational Institutions

**Ms. Nupur Jain[a], Dr. Mahaveer Sain[b]**
[a] PhD Scholar, Maharishi Arvind University, Jaipur, India.

**Article History**: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

**Abstract:** In the recent past, cloud computing technology plays major role in storage of information and big data to ensure privacy and security. The small and medium educational institutions should store their large information with high confidence and effective storage environment. The big data analysis and could computing are used to store information of small and medium educational institutions to store data dynamically and utilizing virtualized resources as a service through the Internet of Things (IoT) by selecting proper service providers (SP). In the proposed system, the selection of SP and the datacenters for small and medium educational institutions are proposed to provide privacy and security of large amount of data. The SP should provide online social network of data, information, 2D images and video of educational institutions for guarantee security of big data. The SP may have laminations in cloud space and giving space to the institutions to keep store higher size of data. The efficient and accurate data, image and video compression algorithm is proposed to minimize the data size to keep store in the cloud to occupy memory space less. The data to be stored through data centers are connected with other SP to minimize the payment cost of educational institutions. This criterion is achieved by applying Modified Genetic Algorithm (MGA) to optimize SP selection among various SP's by achieving bandwidth allocation, maximum throughput, less error and high storage size. The experimental results show that the proposed algorithm is highly efficient and accurate for selection of SP among other providers**.**

**Keywords:** IoT, Big data, SP, Compression, MGA, Cloud computing

## 1. Introduction

The cloud computing has become a significant paradigm for bringing the benefits of cloud computing to for storing big data of small and medium educational institutions. Service availability along with performance enhancement and energy efficiency are primary targets in cloud computing. The offloading framework also called mCloud was used that consists of social network, nearby cloudlets and public cloud services, to improve the performance and availability of the cloud services. The effect of the context (e.g. network conditions) on offloading decisions is studied by proposing a context-aware offloading decision algorithm aiming to provide code offloading decisions at runtime on selecting medium and appropriate cloud resources for offloading. Also investigate failure detection and recovery policies for our mCloud system. It explains in details the design and implementation of the mCloud prototype framework. Real experiments are conducted on the implemented system to evaluate the performance of the algorithm. The system and embedded decision algorithm are able to provide decisions on selecting wireless medium and cloud resources based on different context of the mobile devices, and achieve significant reduction on make-span and energy, with the improved service availability when compared with existing offloading schemes [1]. The storage services of cloud environment are replacing the traditional Content Delivery Networks for more reliability and easy availability of contents to users such as small and medium educational institutions. Most research in Content Delivery Networks mainly focuses on delivering contents to the users with less latency and traffic cost. Apart from this the overall cost incurred for the content providers in cases of bandwidth and storage should also be taken into consideration. Most existing Content Delivery Networks focus only on the bandwidth and in some cases latency. The novel Content Delivery Model was proposed that makes use of a Genetic Optimization Algorithm (GOA) combined with an efficient storage model that can achieve better content placement and delivery in Cloud based Content Delivery Networks. The proposed approach updates itself dynamically to avoid unwanted use of storage that achieves a much better placement of contents thus reducing the storage cost [2]. A secure searchable encryption scheme is presented to enable searching of encrypted user data in the cloud. The scheme simultaneously supports fuzzy keyword searching and matched results ranking, which are two important factors in facilitating practical searchable encryption. A chaotic fuzzy transformation method is proposed to support secure fuzzy keyword indexing, storage and query. A secure posting list is also created to rank the matched results while maintaining the privacy and confidentiality of the user data, and saving the resources of the user mobile devices [3]. The system called SDCon is proposed as a practical platform developed on OpenStack and OpenDaylight to provide integrated manageability for both resources in cloud infrastructure. The platform can perform VM placement and storage, network flow scheduling and bandwidth allocation, real-time monitoring of computing and networking resources, and measuring power usage of the infrastructure with a single platform. Also it proposes a network topology aware VM placement algorithm for heterogeneous resource configuration (TOPO-Het) that consolidates the connected VMs into closely connected compute nodes to reduce the overall network traffic. The algorithm is proposed to evaluate on SDCon and compared with the results from the state-of-the-art baselines [4]. With Service Oriented systems, it moved into a world in which a software application may delegate part of its functionality to already existing software services run by external organizations. Recent advances in Cloud computing are pushing vitality even further: users can access third party software components, hardware physical resources or full application stacks that support execution and automatic management of Cloud based applications, and pay only for the resources they use. Cloud computing is growing daily, providing a vibrant technical environment where innovative solutions and services can be created. The Cloud promises the capability for cheap and flexible services for end-users and allows small organizations and individuals to host and offer world-scale services, themselves [5]. Receiver heterogeneity is increasing in addition due to antenna heterogeneity. As an instance, with the emergence of IEEE 802.11nand IEEE 802.11ac standards, the numbers of antennas at the receivers can vary from 1 to many for multi input and multi output communication through cloud [6]. Multicasting to a set of heterogeneous receivers is tough due to the fact we need to make sure not best each receiver receives video with reasonable great but additionally the receivers with higher channel or more antennas enjoy higher performance rather than being bottlenecked by the weakest receiver [7]. Compressive sensing (CS) is one of the viable information compression techniques where few estimations of an inadequate sign are required to have an accurate recuperation. In Compressive detecting system, the information is procured and packed simultaneously. CS idea permits catching just less inadequate and compressible sign which contains just valuable data that recuperates a legitimate sign. The image compression is performed with compressive detecting idea. The distinctive premise lattices and detecting networks are viewed as which fulfill the Restricted Isometric Property (RIP) and Independent and Identically Distributed (IID) [9]. For the enormous number of the image information created by the sensor hubs in remote mixed media sensor systems (WMSNs), the decrease of the sensor information and vitality proficient transmission of this information are the most sampling issue. Compacted detecting based image compression gives the sensational decrease of image sampling rates, vitality utilization for WMSNs information accumulation and transmission. For better reasonable for continuous detecting of image and lessen the detecting grid size, square compacted detecting technique obtain and procedure image in a square by-square way, by a similar administrator. As of late, nonlocal sparsity has been confirm to improve the reproduction of image subtleties in different compacted detecting thinks about. In this work, in light of square compacted detecting, both the nearby sparsity of the entire image and nonlocal sparsity of comparable patches are incorporated into a packed detecting recuperation system. Furthermore, the nonlocal sparsity of images is portrayed by the low-position estimate. For better estimation of the rank capacity, the non-arched low-position regularization in particular Schatten p-standard minimization is applied for square compacted detecting recuperation [10].

## 2. Related Work

Jian Li et. al [11] has introduced the minimum storage regeneration (MSR) point based distributed storage schemes that can achieve optimal storage efficiency and storage capacity of big data of small and medium educational institutions. Moreover, our scheme can detect and correct malicious nodes with much higher error correction efficiency. For online computing, we investigate outsourcing of general computational problems and propose efficient Cost-Aware Secure Outsourcing (CASO) schemes for problem transformation and outsourcing so that the cloud is unable to learn any key information from the transformed problem. Arjun Kumar et. al [12] has presented a method to build a trusted computing environment by providing a secure platform in a Cloud computing system. The proposed method allows users to store data safely and efficiently in the Cloud. It solves the problem of handling big data and security issues using encryption and compression technique while uploading data to the Cloud storage. Xiao Chun Yin et. al [13] has proposed a scheme to securely store and access of data via internet. We have used ECC based PKI for certificate procedure because the use of ECC significantly reduces the computation cost, message size and transmission overhead over RSA based PKI as 160-bit key size in ECC provides comparable security with 1024-bit key in RSA. We have designed Secured Cloud Storage Framework (SCSF). In this framework, users not only can securely store and access data in cloud but also can share data with multiple users through the unsecure internet in a secured way. Shenling Liu et. al [14] has presented architecture of a Security Cloud Storage Gateway, which provides implicit storage security to improve security and availability of Cloud Computing. We designed efficient data fragment algorithms based on IDA (Information Dispersal Algorithm), with which we can partition a file into a designated number of pieces and reform by part of the pieces efficiently. We also designed a Cloud Storage Application Programming Interface (CSAPI) for users, a method for accessing to and utilizing a multi-Cloud storage, ignoring difference, between various cloud servers. Jiabin Deng et. al [15] has proposed the use of Power-law Distributions and Improved Cubic Spline Interpolation for multi-perspective analysis of shareware download frequency. The tasks include data mining the usage patterns and to build a mathematical model. Through analysis and checks, in accordance with changes to usage requirements, our proposed methods will intelligently adjust the data redundancy of cloud storage. Thus, storage resources are fine tuned and storage efficiency is greatly enhanced. Quanlu Zhang et. al [16] has proposed a novel data hosting scheme (named CHARM) which integrates two key functions desired. The first is selecting several suitable clouds and an appropriate redundancy strategy to store data with minimized monetary cost and guaranteed availability. The second is triggering a transition process to re-distribute data according to the variations of data access pattern and pricing of clouds. It evaluates the performance of CHARM using both trace-driven simulations and prototype experiments. The CHARM not only saves around 20% of monetary cost but also exhibits sound adaptability to data and price adjustments. Haiying Shen et. al [17] has proposed a multi-cloud Economical and SLO-guaranteed Storage Service (ES3), which determines data allocation and resource reservation schedules with payment cost minimization and SLO guarantee. ES3 incorporates (1) a coordinated data allocation and resource reservation method, which allocates each data item to a datacenter and determines the resource reservation amount on datacenters by leveraging all the pricing policies; (2) a genetic algorithm based data allocation adjustment method, which reduce data Get/Put rate variance in each datacenter to maximize the reservation benefit. It also proposes several algorithms to enhance the cost efficient and SLO guarantee performance of ES3 including i) dynamic request redirection, ii) grouped Gets for cost reduction, iii) lazy update for cost-efficient Puts, and iv) concurrent requests for rigid Get SLO guarantee. Our trace-driven experiments on a supercomputing cluster and on real clouds (i.e., Amazon S3, Windows Azure Storage and Google Cloud Storage) show the superior performance of ES3 in payment cost minimization and SLO guarantee in comparison with previous methods. J.M. Shapiroet. al [18] has presented frame compression evaluation could be belongings created arranged by implication, yielding a completely inserted code. The inserted code speaks to an understanding of paired choices that recognize a frame from the "invalid" frame. Utilizing an inserted coding computation, an encoder can end the encoding anytime accordingly permitting an objective rate or target twisting measurement to be met exactly. Likewise, given a part stream, the decoder can stop unscrambling anytime in the bit stream and still create the very same frame that would have been encoded at the bit rate relating to the edited part stream. Notwithstanding creating a completely installed bit stream, the EZW dependably delivers compression results that are focused with for all intents and purposes all known compression calculations on standard test images. Said A, Pearlman W An et. al [19] has presented convincing and computationally basic procedure for frame compression. We offer an optional explanation of the standards of its activity, with the goal that the purposes behind its smart exhibition can be better comprehended. These standards are incomplete demands by extent with a set allocating arranging computation, requested fraction data transmission, and misuse of self-likeness diagonally over different sizes of a frame wavelet transform. Guesmi Ret. al [20] has proposed a shading frame encryption plot utilizing one-time keys dependent on hybrid administrator, disorder and the Secure Hash Algorithm (SHA-2). The (SHA-2) is utilized to create a 256-part hash a motivation from both the plain-frame and the unknown hash keys to make the key stream change in every encryption process. The SHA-2 worth is utilized to make 3 starting estimations of the turbulent system.

**3. System Model**

In this work, we consider the cloud storage services offered in a multi-cloud environment for big data based small and medium educational institutions, which involves two types of entities: i) Users, who store a large number of encrypted files in multiple clouds and execute keyword-based queries to access and manipulate their stored files; ii) Service Providers (SPs), who possess storage and computation resources, are willing to cooperatively store and manage the users' files. We focus on searchability of encrypted data, stored by users in one or many multi-cloud service providers. Informally, searchability (of encrypted data) refers to the ability of end users to retrieve encrypted files without having the SP to decrypt it. These searches are typically carried out using keywords, which the client uses to locate the desired files. We formalize this notion of general keyword search problem on plain files in multi-cloud environment. The cloud data is collected through IoT at small and medium educational institutions at the server location. Given a user, let f be a collection of user files stored in a number of SPs located in multiple clouds. A keyword search query Q(wq) issued by the user retrieves these files (from the SPs), that contain the query keyword wq. Though many works focus on finding the minimum resource to support the workload to reduce cloud storage cost in a single SP, there are few works that studied cloud storage cost optimization across multiple SPs with different prices. SPANStore aims to minimize the cloud storage cost while satisfying the latency and failure requirement across multiple SPs. However, it neglects both the resource reservation pricing model and the datacenter capacity limits for serving Get/Put requests. A datacenter's Get/Put capacity is represented by the Get/Put rate (i.e., the number of Gets/Puts in a unit time period) it can handle. Reserving resources in advance can save significant payment cost for customers and capacity limit is critical for guaranteeing SLOs since datacenter network overload occurs frequently. The integer program used to create a data allocation becomes NP-hard, if it is modified with capacity-awareness, which however cannot be easily resolved. As far as we know, our work is the first that provides minimum-cost cloud storage service across multiple SPs with the consideration of resource reservation and datacenter capacity limits.

We have modeled the cost minimization problem under multiple constraints using integer programming.

We introduce a heuristic solution including:

(1) A dominant-cost based data allocation algorithm, which finds the dominant cost (Storage, Get or Put) of each data item and allocates it to the datacenter with the minimum unit price of this dominant cost to reduce cost in the pay-as-you-go manner.

(2) An optimal resource reservation algorithm, which maximizes the saved payment cost by reservation from the pay-as-you-go payment while avoiding over reservation.

We conduct extensive trace-driven experiments on a supercomputing cluster and real clouds (i.e., Amazon S3, Windows Azure Storage and Google Cloud Storage) to show the effectiveness and efficiency of our system in cost minimization, SLO compliance and system overhead in comparison with previous systems. The figure 1 shows the architecture of multi-cloud collaboration interfacing between various service providers.

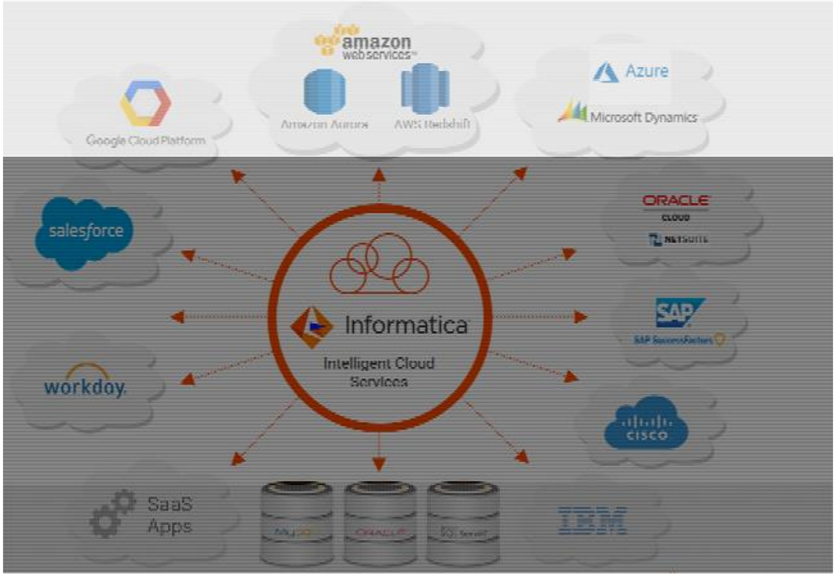


Figure 1: Multi-Cloud architecture diagram

**4. Problem Statement**

We have called a datacenter that operates an educational institutions application datacenter of this customer. According to the operations of a customer's clients, the customer datacenter generates read/write requests to a storage datacenter storing the requested data. A customer may have multiple customer datacenters (denoted by Dc). We use dci∈ Dc to denote the ith customer datacenter of the customer. We use Ds to denote all datacenters provided by all cloud providers and use dpj ∈ Ds to denote storage datacenter j. A client's Put/Get request is forwarded from a customer datacenter to the storage datacenter of the requested data. The cloud storage customers need data request (Puts/Gets) deadlines for their applications, and need to avoid the data request failures. One type of SLO specifies the Get/Put bounded latency and the percentage of requests obeying the deadline. Another type of SLO guarantees the data availability in the form of a service probability by ensuring a certain number of replicas in different locations. Data storage and request Allocation and resource Reservation (DAR) considers both types to form its SLO and can adapt to either type easily. This SLO specifies the deadlines for the Get/Put requests (Lg and Lp), the maximum allowed percentage of data Get/Put operations beyond the deadlines (_g and _p), and the minimum number of replicas (denoted by β) among storage datacenter. For a customer datacenter's Get request, any storage datacenter holding the requested data (i.e., replica datacenter) can serve this request. A cloud storage system usually specifies the request serving ratio for each replica datacenter of a data item during billing period tk (e.g., one month). The SPs charge the customers by the usage of three different types of resources: the storage measured by the data size stored in a specific region, the data transfer to other datacenters operated by the same or other SPs, and the number of Get/Put operations on the data. The storage and data transfer are charged in the pay-as-you-go manner based on the unit price. The Get/Put operations are charged in the manners of both pay-as-you-go and reservation. In the reservation manner, the customer specifies and prepays the number of Puts/Gets per reservation period T (e.g., one year). The unit price for the reserved usage is much cheaper than the unit price of the pay-as-you-go manner (by a specific percentage). For simplicity, we assume all datacenters have comparable price discounts for reservation. That is, if a datacenter has a low unit price in the payasyou-go manner, it also has a relatively low price in the reservation manner. The amount of overhang of the reserved usage is charged by the pay-as-you-go manner. Therefore, the payment cost can be minimized by increasing the amount of Gets/Puts charged by reservation and reducing the amount of Gets/Puts for over reservation, which reserves more Gets/Puts than actual usage. For a customer, DAR aims to find a schedule that allocates each data item to a number of selected datacenters, allocates request serving ratios to these datacenters and determines reservation in order to guarantee the SLO and minimize the payment cost of the customer. In the following, we formulate this problem using integer programming. We first set up the objective of payment minimization. Finally, we formulate the problem with the object and constraints.

Cost minimization objective

We aim to minimize the total payment cost for a customer (denoted as Ct). It is calculated as

Ct = Cs + Cc + Cg + Cp, (1)

where Cs, Cc, Cg and Cp are the total Storage, Transfer, Get and Put cost during entire reservation time T, respectively. DAR has two steps. First, its dominant-cost based data allocation algorithm conducts storage and request allocation scheduling that leads to the lowest total payment only in the pay-as-you-go manner. Second, its optimal resource reservation algorithm makes a reservation in each used storage datacenter to maximally reduce the total payment.

•Dominant-cost based data allocation algorithm. To reduce the total payment in the pay-as-you-go manner as much as possible, DAR tries to reduce the payment for each data item. Specifically, it finds the dominant cost

(Storage, Get or Put) of each data item and allocates it to the datacenter with the minimum unit price of this dominant cost.

•Optimal resource reservation algorithm. It is a challenge to maximize the saved payment cost by reservation from the pay-as-you-go payment while avoiding over reservation. To handle this challenge, through theoretical analysis, we find the optimal reservation amount, which avoids both over reservation and under reservation as much as possible. The main components of the Cloud framework belong to two parts: client part and cloud server part. As depicted in Figure 1, there are five main components on the client side: Context Monitor, Decision Module, Task Manager, Communication Manager and Failure Recovery. On the cloud server side, main components include Communication Handler, Task Manager and corresponding wireless sensor cloud infrastructures.

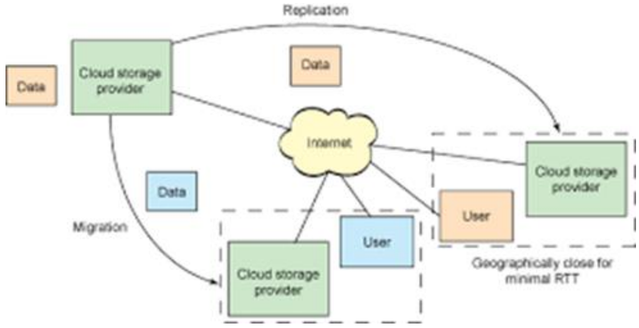


Figure 2: Cloud storage

Modified Genetic Algorithm (MGA)

To more effectively identify hot data and minimize the amount of data storage, the data are divided in the form of partition in each storage node, that is, a data storage node can be divided into multiple partitions. Zoning is the basic unit of data storage and load monitoring. Before using the data storage algorithm, we should classify the storage nodes into collection of storage nodes In_set and Out_set separately according to the relations between normalized loading value and 1/n. In_set contains the node tha load value below 1/n and Out_set contains the nodes that the load value higher than 1/n. Target of Cloud data storage is to achieve a system load balance between nodes in a cloud environment, that is, it should migrate a part of the data of Out_set node to the node of In_set. For such problems, particle swarm optimization algorithm calculates the average shortest time of each data partition migrated between nodes. The algorithm can control the global and local data storage and ultimately it achieve optimize system resources. The whole process should follow these principles:

a) Principle of proximity, the group can perform simple calculation of space or time, the selection of node in In_set is based on the principle of proximity.

b) When the system meets the threshold setting of the unbalanced degree, we use MGA algorithm for each node of out_set at the same time.

Algorithm Steps

Step1: set the algorithm parameters: size of population and dimensions, maximum number of iterations Tmax or the expected detected entropy, the inertia weight Wmax, Wmix and the optimal solution set.

Step2: Initialize each population, calculate the fitness value of each particle Fij (x1, x2, x3).

Step3: Particles conduct migration in the system in accordance with each particle's value Fi (x1, x2, x3) in which following the principle: follow the migration process, the particles move to immigration node with the minimal cost (Fi (x1, x2, x3) value is the smallest node). The fames value of each particle compare with Pbest, if the current position is better, it will be the best position Pbest;

Step4: For each particle, compare its fitness value and the best position it passed Pbest .If the former is better, take it as current best position.

Step5: Obtain particle's new speed and position based on the formula of iterations. When the load value of immigration node is equal to the total load value /number of nodes of the threshold value of system, then the node is automatically removed.

Step6: the termination condition: the number of iterations reaches the maximum number of iterations set or achieve set detection value (entropy) of load balancing

Pseudo code of MGA

// Function: pseudo-code of MGA
// Note: this example problem aims for minimum fitness for the purpose
// Parameters: N as the population size, that is,the number of partitions
For each particle i
Initialize velocity Vi and position Xi for particle i
Evaluate particle i pBest and gBest

End for
gBest=min {pBest}
while not stop
for I to N
Update the velocity and position of particle i
Evaluate particle i
If fit(Xi)<fit(pBest)
pBest=Xi;
If fit(pBest)<fit(gBest)
gBest=pBest;
end for
end while
print gBest
end

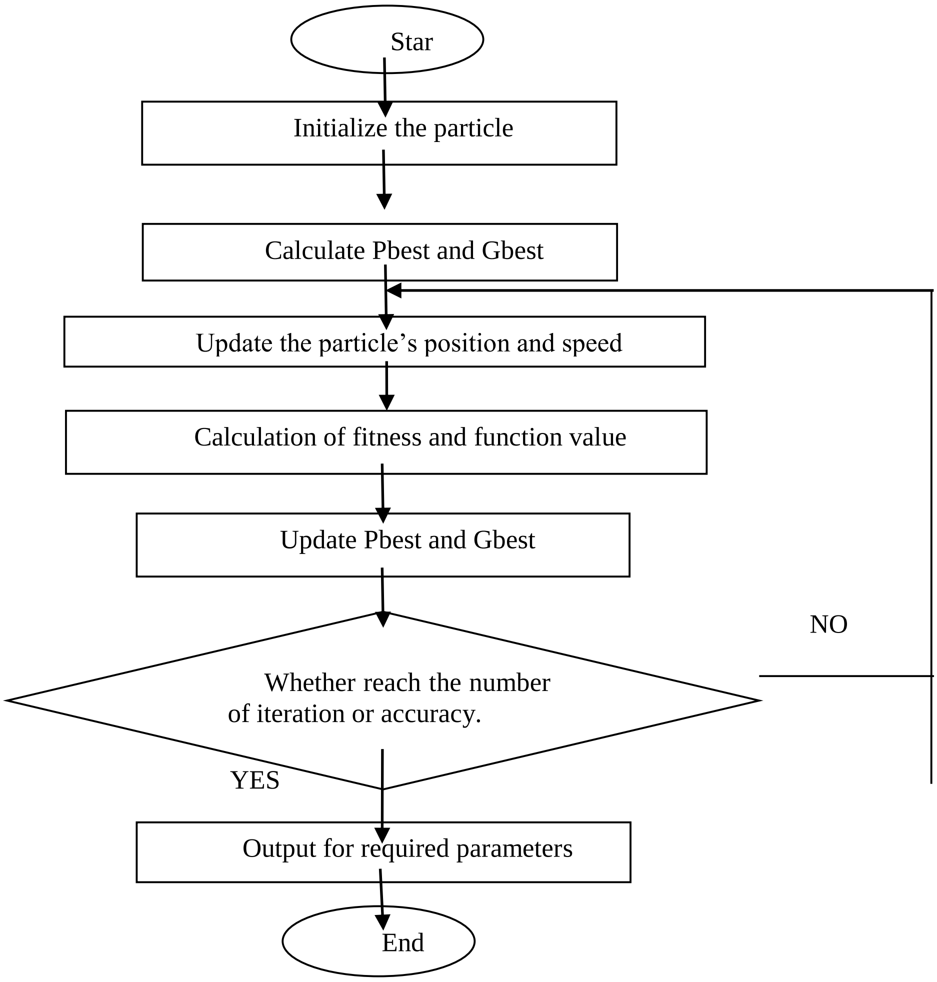


Figure 3: MGA algorithm flow chart

COST Minimization ALGORITHM for SP selection using MGA

We first introduce the system model and the problem formulation. Then we present the details of our cost estimation model and the context-aware offloading algorithm. The system considers a heterogeneous cloud environment, consisting of a ad-hoc cloud, the nearby cloudlets and remote public cloud. There are a set of device users that run applications seeking opportunities to offload tasks to the cloud infrastructure. Having presented the models of the system, we formulate the decision making problem as to find a solution of selecting where to execute the task and how to offload so that the overall execution time and energy consumption is the lowest among all the cloud resources in the wireless sensor cloud infrastructure based on the current context of the client device. Specifically, for a given a set of n tasks T, a set of cloud a set of cloudlets CL, and a wireless sensor ad-hoc cloud with h wireless sensor devices M, then the overall cost of executing a set of n tasks. Having shown how to estimate the task execution time and energy consumption with the cost models, we present the algorithm in this section. In order to obtain the lowest execution cost for the off-loadable tasks under the context, the context-aware decision algorithm considers a set of context parameters, multiple wireless medium and wireless sensor cloud infrastructure resources to decide when it is beneficial to offload, which wireless medium is used for offloading and which resources to use as the offloading location. Most existing offloading frameworks in the literature only consider network speed and energy consumption when they make offloading decisions. Unlike those, cloud focuses on utilizing multiple types of wireless sensor cloud resources (i.e. cloud, cloudlet and wireless sensor ad-hoc cloud) and wireless mediums based on device context to improve the offloading service availability and performance. Multiple criteria regarding device context including resource availability, wireless medium availability, network congestion, cost, energy consumption, etc. have been considered for making offloading decisions in cloud. Therefore, we need a multi criteria decision making approach (MCDM) in the proposed framework. Among MCDMs, we apply Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) for wireless medium selection considering abovementioned criteria. TOPSIS offers lightweight processing and shorter response time comparing to other MCDMs. It helps reduce the overhead of the proposed offloading decision making algorithm, considering it is running on wireless sensor devices. Moreover, TOPSIS can be easily modified to consider more criteria if necessary, and the complexity remains the same regardless of the number of criteria. In cloud, the decision making algorithm considers six criteria related to performance when selecting the wireless interface:

1. energy cost of the channel,
2. the link speed of the channel,
3. the availability of the interface,
4. monetary cost (i.e. cost when using wireless sensor data),
5. the congestion level of the channel (RTT),
6. and the link quality of the channel (signal strength).

Note that for the monetary cost, the algorithm only considers the cost generated by using the data.

## 5. Results and Discussion

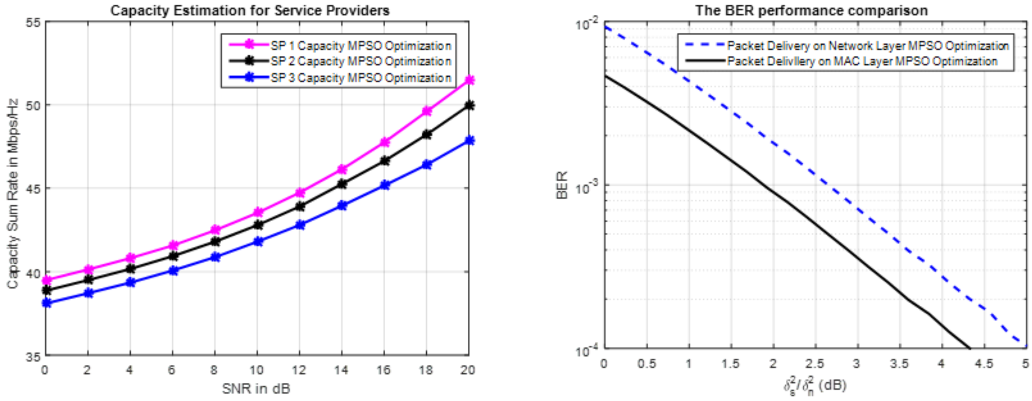


Figure 4: (a) Capacity estimation for various service providers, (b) BER performance for packet delivery of OSI layers for cloud storage

The figure 4 (a) shows the capacity estimation for various service providers. The water filling algorithm is used to estimate the capacity in terms of Mbps/Hz for each service provider using MGA algorithm. As shown in the figure, the signal to noise ratio increases the capacity of service provider increases. The figure 4 (b) shows the bit error rate reduction during the packet delivery ratio using MGA algorithm.

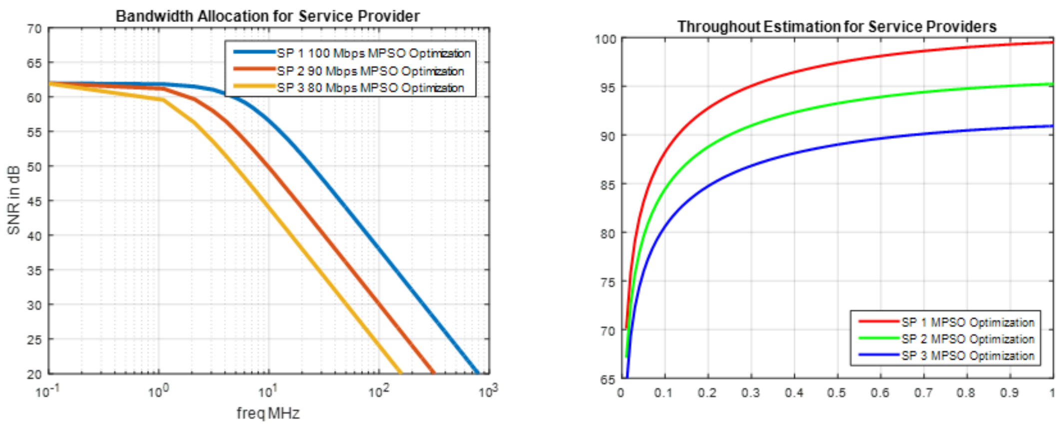


Figure 5: (a) Bandwidth allocation for service providers using MGA optimization algorithm, (b) Throughput estimation using MGA Optimization algorithm

The figure 5 (b) shows throughput maximization for service providers with MGA optimization. The throughput rate is high for cloud system based on its performance. Obtaining signal strength metrics for wireless connections is not easy. Within the past, once coping with 3G technologies, there has continuously been an unambiguous and easily out there metric-the Received Signal Strength Indicator (RSSI). This metric was helpful once neighbor cells could not share frequencies, and then it had been attainable to directly attribute the strength of a system contained inside one frequency to one cell. As digital modulation technologies have progressed, and network operators have developed a lot of complex and overlapping cell topologies, this restriction has been upraised in order that this metric is of immensely decreased relevancy.


Figure 6: Deployment of service providers using MGA optimization algorithm

Figure 6 shows the deployment of service provider’s estimation using MGA optimization algorithm. The deployment of service providers are considered for two parameters such as Load balancing and signal to noise ratio. The service provider 1 is considered as the optimized version of deployment.

TABLE I. STORAGE PRICE OF CLOUD COMPANY A

| **Name** | **CPU(s)** | **Memory(GB)** | **Price($/Hours)** | | |
|---|---|---|---|---|---|
| | | | **Place 1** | **Place 2** | **Place 3** |
| m1.small | 1 | 1.7 | 0.085 | 0.095 | 0.095 |
| m1.large | 4 | 7.5 | 0.34 | 0.38 | 0.38 |
| m1.xlarge | 8 | 1.5 | 0.68 | 0.76 | 0.76 |
| c1.medium | 5 | 1.7 | 0.17 | 0.19 | 0.19 |
| c1.xlarge | 20 | 7 | 0.68 | 0.76 | 0.76 |
| m2.xlarge | 6.5 | 7 | 0.50 | 0.57 | 0.57 |
| m2.xlarge | 13 | 34.2 | 1.00 | 1.14 | 1.14 |
| m2.4large | 26 | 68.4 | 2.00 | 2.28 | 2.28 |

**TABLE II. STORAGE PRICE OF CLOUD COMPANY B**

| **Name** | **CPU(s)** | **Memory(GB)** | **Price($/Hours)** |
|---|---|---|---|
| | | | Place 3 |
| X-small | 0.5 | 0.5 | 0.095 |
| Small | 1 | 1 | 0.19 |
| Medium | 2 | 2 | 0.38 |
| Large | 4 | 4 | 0.76 |
| X-large | 8 | 8 | 1.52 |

| | | | |
|---|---|---|---|
| XX-large | 16 | 16 | 3.04 |

**TABLE III. TRANSMISSION PRICE OF COMPANY A AND COMPANY B**

| Cloud service providers | Price ($/GB) | |
|---|---|---|
| | Data incoming | Data outgoing |
| Company A(all) | 0.10 | 0.15 |
| Company B | 0.00 | 0.29 |

The various parameters are calculated and tabulated in table I, II and III. The transmission price of the company A and B are compared in terms data incoming and data outgoing. The storage price of company A and B are estimated and tabulated.

**6. Conclusion**

In this paper, the cloud computing for the multi data transmission, reception and storage is computed. The small and medium educational institutions are beneficial due to big data technology of their higher rate of storage with high security and privacy. The SP is selected based on the confidentiality and other parameters such as efficiency and accuracy of storing information. The service provider for cloud data storage is optimized using MGA optimization algorithm. The various parameters such as efficiency, bandwidth allocation, throughput and error rate minimization for packet delivery are calculated and proven that the service provider optimization.

**References**

1. Bowen Zhou, Amir Vahid Dastjerdi, Rodrigo N. Calheiros, Satish Narayana Srirama, and Rajkumar Buyya, “mCloud: A Context-aware Offloading Framework for Heterogeneous Mobile Cloud”,IEEE 2015.
2. S.Sajithabanu , S.R.Balasundaram , “ Cloud based Content Delivery Network using Genetic Optimization Algorithm for Storage Cost ”,IEEE 2017.
3. Abir Awad, Adrian Matthews, Yuansong Qiao, Brian Lee, “Chaotic Searchable Encryption for Mobile Cloud Storage”, IEEE Transactions on cloud computing.
4. Jungmin Son, Student Member, IEEE, and Rajkumar Buyya, “SDCon: Integrated Control Platform for Software-Defined Clouds”, IEEE Transactions On Parallel And Distributed System, 2017 .
5. Danilo Ardagna, “Cloud and Multi-Cloud Computing: Current Challenges and Future Applications”, 2015 IEEE/ACM 7th International Workshop on Principles of Engineering Service-Oriented and Cloud Systems.
6. J. Weir and W. Yan, “Visual cryptography and its applicatins,”London, UK: Ventus Publishing, 2012.
7. C. Blundo and A. D. Santis, “Visual cryptography schemes withperfect reconstruction of black pixels”, Computers & Graphics, vol.22, no. 4, pp. 449□455, 1998.
8. Po-Wen Chi and Chin-Laung Lei , “Audit-Free Cloud Storage via Deniable Attribute-based Encryption”, IEEE Transactions on cloud computing,2015 .
9. Amit M. Joshi , Chitrakant Sahu , M. Ravikumar , Samar Ansari, “Hardware implementation of compressive sensing for image compression”, 10.1109/TENCON.2017.8228060, IEEE 2017.
10. Yan Zhang , Jichang Guo , Chongyi Li, “Compressive sensing in wireless multimedia sensor networks based on low-rank approximation”, 10.1109/ICC.2017.7997066, IEEE 2017.
11. Jian Li , Kai Zhou , Jian Ren, “Security and efficiency trade-offs for cloud computing and storage”, 10.1109/RWEEK.2015.7287434. IEEE 2016.
12. Arjun Kumar , HoonJae Lee , Rajeev Pratap Singh, “Efficient and secure Cloud storage for handling big data”, 2012 6th International Conference on New Trends in Information Science, Service Science and Data Mining (ISSDM2012), IEEE 2013
13. Xiao Chun Yin , Zeng Guang Liu , Hoon Jae Lee, “An efficient and secured data storage scheme in cloud computing using ECC-based PKI”, 10.1109/ICACT.2014.6779015, IEEE 2014
14. Shenling Liu , Chunyuan Zhang , Le Bo, “Improve security and availability for cloud storage”, 10.1109/CCIS.2016.7790288, IEEE 2016.

15. Jiabin Deng , JuanLi Hu , Anthony Chak Ming Liu , Juebo Wu, “Research and Application of Cloud Storage”, 10.1109/IWISA.2010.5473373, IEEE 2016.
16. Quanlu Zhang, Shenglong Li, Zhenhua Liy, Yuanjian Xingz, Zhi Yang, and Yafei Dai, “CHARM: A Cost-efficient Multi-cloud Data Hosting Scheme with High Availability”, IEEE Transactions on Cloud Computing 2015.
17. Haiying Shen, Guoxin Liu and Haoyu Wang, “An Economical and SLO-Guaranteed Cloud Storage Service across Multiple Cloud Service Providers”, IEEE 2017.
18. Shapiro J M. Embedded image coding usingzerotrees of wavelet coefficients [J]. IEEE Transactions on signal processing, 1993, 41(12):3445-3462.
19. Said A,Pearlman W A. A new,fast,and efficient image codec based on set partitioning in hierarchical trees[J]. IEEE Trans.on CircuitsSyst.Video Technol,1996,6(3):243-250.
20. Guesmi R, Farah M A B, Kachouri A, et al. Hash key-based image encryption using cross over operator and chaos[J]. Multimedia tools and applications, 2016, 75(8): 4753-4769.