ISSN: 0974-6471, Vol. 13, No. (2-3) 2020, Pg. 82-90

Oriental Journal of Computer Science and Technology

www.computerscijournal.org

Maintaining the Data Integrity and Data Replication in Cloud using Modified Genetic Algorithm (Mga) and Greedy Search Algorithm (Gsa)

M. RAMANAN* and J. AROCKIA STEPHEN RAJ

Department of Physical Sciences and IT,Agricultural Engineering College and Research Institute, Tamil Nadu Agricultural University, Coimbatore-641003, Tamil Nadu, India.

Abstract

Cloud computing is now most widely used platform to store and access data globally. Data Integrity is an important aspect which ensures the quality of the data stored in the cloud and maintaining integrity is a challenging task. Data Replication is the key factor which maintains replication of data stored in the cloud. The integrity of data will be affected by various factors like intruders, data corruption and suspicious entities posing threat to data in cloud. The prevailing data replication methods are not able to handle dynamically changing large volume of data and obtaining additional storage resources for creation of replicas. Owing to several security threats existing incloud, an effective mechanism is needed to audit the integrity of data. In this paper, approaches for maintaining the integrity of data and data replication are proposed to improve the service quality of the Cloud Provider. The MGA and GSA approaches are based on deterministic approach for finding near optimal solution. Both MGA and GSA methods reduces the lag in managing the remote data by storing the data closer to the applications.

Article History

Received: 28 October 2020

Accepted: 24 December 2020

Keywords

Cloud Computing;
Data Integrity;
Data Integrity;
Data Replication;
Modified Genetic Algorithm (Mga);
Greedy Search Algorithm (Gsa).

Introduction

Cloud Computing provides a platform for delivery of computing services like storage, networking, analytics etc., over the internet.[1] Such advancement in computing technologies has made major IT organizations start to move their data towards cloud environment. The Cloud platform offers a light weight flexible way of storing and accessing data with innovative newer technologies and resources. This will facilitate to reduce the operational cost and to executethe computational needs in a high end infrastructure effectively. The major challenges encountered by the cloud storage environment are the problem of maintaining the integrity of data and

CONTACT M. Ramanan pmramanan@gmail.com Department of Physical Sciences and IT,Agricultural Engineering College and Research Institute, Tamil Nadu Agricultural University, Coimbatore-641003, Tamil Nadu, India.

Doi: http://dx.doi.org/10.13005/ojcst13.0203.04

effectively creating and managing the replicas of data.

The integrity of data is considered to be the maintenance of the intactness of data.[2] During various operations like accessing the data, data retrieval, data replication, data storage etc., the integrity of data should be maintained. The data can be accessed, modified or updated only if a suitable operation is authorized. Data integrity may be hampered at the storage level due to various factors. The media types that can result in a data corruption will be the bit rot. Bit rot results in data duplications, metadata corruption and controller failures. This is a very critical factor which ruins the integrity by altering the bits of data for any reason. For example, the text file integrity may get affected by including a single space character within the file. In these cases, making an alteration to some words can render the situation risky.

Cloud providers offers a reliable environment to the users for storing and accessing the data in the cloud, but integrity of the data is hard to maintain due to human errors and software failures. Many research works are being carried out to maintain data integrity. These mechanisms uses signature based authentication mechanisms using a Third Party Actuary (TPA). This TPA will be able to check the data integrity by accessing only a part of the data.[5] The study of data storage and replication in cloud environment provides an unrivalled opportunity to know about various storage methods involved in cloud. The main objective of a good cloud storage is to identify any abnormal activity and resolve the problems accordingly. There are several existing solutions for cloud security frame work and many approach for vulner abilities coverage and cost optimization, which mitigates identified set of vulner abilities using selected set of techniques by minimizing cost and maximizing coverage.[11] However, there are still number of key challenges yet tobe solved in selecting the appropriate methodologies for effective cloud storage and replication

In this paper, the MGA and GSA approaches are based on deterministic approach for finding near optimal solution is discussed. Both MGA and GSA methods reduce the lagby storing the data nearer to the services.The MGA method chooses a set of evenly distributed data blocks, applies a fitness function and selects the best nodes for replication. This process will continue untila best solution is obtained. The GSA approach starts with an inceptive solution for a problem and fixes the problem in a step by step manner so as to obtain optimal solution in a least time.[12]

The cloud environment provides high computation power to its users by abstracting the computations involved in background. The abstraction can be implemented as private cloud or public cloud. There are several factorsthat require processing and management of large data with state of art computational standards. Lots of research work is being carried out to improve the computational capabilities (Rashmi Ranjana *et al*., 2015).

The approaches discussed in this paper can be used to reduce the lag in accessing the remote data. By replicating data in various data center, the performance can be increased. The accesses through local databases pose challenges with respect to synchronization of data and many replicas storage. In this work, an optimized data replication algorithm is proposed.

Related Works

A formal evaluation of the probable types of fine grained updates of data has been provided by Liu *et al* ., (2014). A scheme has been suggested that can completely take care of the verified auditing as well as intricate update requests. An enhancement has also been suggested on the basis of the approach; this scheme is known to spontaneously decrease the burdens of communication whenever trivial updates are being verified. It has been proven via theoretical and practical verifications that the scheme can increase both the security as well as the flexibility. It can also sufficiently decrease the overheads for big data schemes involving several small updates like the social media applications as well as business transactions.

For removing the complexity of the certificate management in conventional heuristics for cloud data integrity verification, an Identity-based Cloud Data Integrity Checking (ID-CDIC) protocol has been suggested by Yu *et al*., (2016). Different sized file blocks as well as public auditing can be supported by the proposed concrete construction from Ron Rivest, the Adi Shamir and the Leonard Adleman (RSA) signature. Additionally, a formal security model for ID CDIC can be provided. This can verify the formulation's security. This is under RSA assumption with large public exponents in the random oracle model. The model of the heuristic has been developed for showing the performance of this scheme. It has been shown via implementation outcomes that in real life situations, the suggested ID-CDIC has been extremely practical as well as adaptable.

A case has been taken up by Gaetani *et al*., (2017). This is from the European SUNFISH project. It involves the formulation of cloud federation platform for public sector which is a secure design. This comprises the data integrity requirements of the cloud computing set ups. For adopting the databases based on block chain, there is a need to address the research questions. Firstly, the persisting research questions have been detailed. Then, for addressing these questions, the intrinsic challenges have been addressed. This is followed by outlining the basic formulation of the database based on block chain in cloud computing paradigm.

The replication of data in cloud computing data centers has been explored by Boru *et al*., (2015). This approach is both effective in terms of energy as well as the system band width consumed, unlike the other schemes. Because of decreased communication delays, there is an improvisation in the Quality of Service that has been obtained. The outcomes of analysis from the pervasive simulations as well as the mathematical prototype have proven that there is a tradeoff between the performance as well as energy efficiency. It also guides the formulation of future solutions for replication of data. A suggestion has been made for Optimal Performance and Security (DROPS) by Ali *et al*., (2015). This is for splitting and replicating data. This has a collective approach to tackle both the security and the performance issues. The DROPS scheme splits the file into various parts and this data which is split is replicated over the cloud nodes. Since only a tiny part of the data file is contained in every node, it ensures that even when a successful attack takes place, the data exposed to the attacker will have no critical information. Additionally, the nodes that have the fragmented data are all at a distance of each other using a graph T-coloring so that their placement remains elusive to the attacker. The systems are also relieved from computationally expensive schemes as the DROP technique does not depend on the conventional basis of data as well as security.

Zhang *et al*., (2016) made a proposal of Provable Multiple Replication Data Possession protocol having full dynamics called the MR-DPDP. A genuine structure of the data referred to as the Merkle hash tree is used in MR-DPDP. This comprises grading which can support the spontaneous updates on data, along with the data verification as well as protection. The file blocks of different sizes are supported by the formulation with the RSA signature. This uses the proof of security and also evaluates the performance for showing the MR-DPDP that lesser communication overheads are encountered when the data blocks are being updated and the proof of integrity for several replicas is being verified.

There are several existing solutions for cloud security framework and many approach forvulnerabilities coverage and cost optimization, which mitigates identified set ofvulnerabilities using selected set of techniques by minimizing cost andmaximizing coverage. However, there are still number of key challenges yet tobe solved in selecting the appropriate methodologies for effective cloud storageand replication.

Methodology

In this section, an improved methodology of maintaining data integrity and data replication using the Modified Genetic Algorithm (MGA) and Greedy Search algorithm are discussed.

Modified Genetic Algorithm (MGA)

The problem of data replication is: let $F=\{f_{1,} f_{2'},..., f_{1}\}$ is a file group belonging to a data hub. $B=\{B_{1,} B_{2'},...,$ $B_{1}\}$ is the file blocks of data hub, and $B_{i}=\{b_{i1,} b_{i1'},...,$ $b_{in1}\}$ sub blocks belonging to ith data file f_{1}, which is distributed evenly. A five row relation which is defined as, $b_{k} = (bid_{k'}; bp_{k'}; bs_{k'}; bn_{k'}; bt_{k})$ where $bid_{k'}$ $bp_{k'}$ $bs_{k'}$ $bn_{k'}$ bt_{k} are the sub section identification, request count, size of the section, replica numbers and the last access time respectively (Hussein & Mousa 2012).

When a block b_{k} is requested by user u_{j} for a node $dnd_{i'}$ with a bandwidth bs_{k}/ bst_{k}. The total band width used required to support n users should be less than dbw$_i$, as shown below

$$\sum_{i=0}^{s_i} \frac{bs_k}{dst_i} \le dbw_i$$

Where s_i is the concurrent network sessions peak value for data node, dnd_i, bs_k is the section size of section b$_k$, dst_i is the average service time of data node, dnd_i, dbw$_i$ is the network bandwidth of data node. The section availability of a section b$_k$ is denoted as BA$_k$. P (BA$_k$) is the probability of section b$_k$ in an available state. So the section vailability can be calculated as

$$P(BA_k) = 1-(1-P(ba_k))^{bn_k}$$

$$\hat{P}(BA_k) = (1-P(ba_k))^{bn_k}$$

If the data file is distributed into fixed sections denoted by, B_i={b_{i1}, b_{i2},..., b_{in1}} which are distributed on different data nodes. N_i={bn_{i1}, bn_{i2},..., bn_{in1}} is the set of the numbers of replicas of the blocks of. B_i. The availability and unavailability of data file f_i is given as:

$$P(FA_i) = (1-(1-P(ba_i))^{bn_i})^{n_i}$$

The various stages that a data is processed in MGA algorithm is shown in Figure 1.

The Steps of MGA Algorithm are Iterated

Initial Population: A set of randomly selected chromosomes are considered as initial population set in genetic algorithm. Every answer in the group of nodes is referred to each single node. Every node is represented as a chromosome for generating different operations. From adistinct group, a node will be selected, and operations are applied on them to form the next generation. A specific criterion is used for mating the chromosomes.

Fitness Function

The efficiency of a node depends on the fitness value. It is the measure of the supremacy of node in the group. Fitness (F) is calculated as the shortest distance between two nodes as shown in

F=1/min(E$_d$ (m,n))

The Probability of selecting best nodes using the fitness function is shown in 4.6.

$$P_j = \frac{F_j}{\sum_{t=1}^{N} F_t}$$

Here Pj is the probability of choosing jth node, N is number of nodes, Fj is the probability of jth node to be fit. The fitness value shows the efficiency of anode in the group. Therefore, the individual node will survive according to the fitness value. Hence, this function is the key factor in this algorithm.

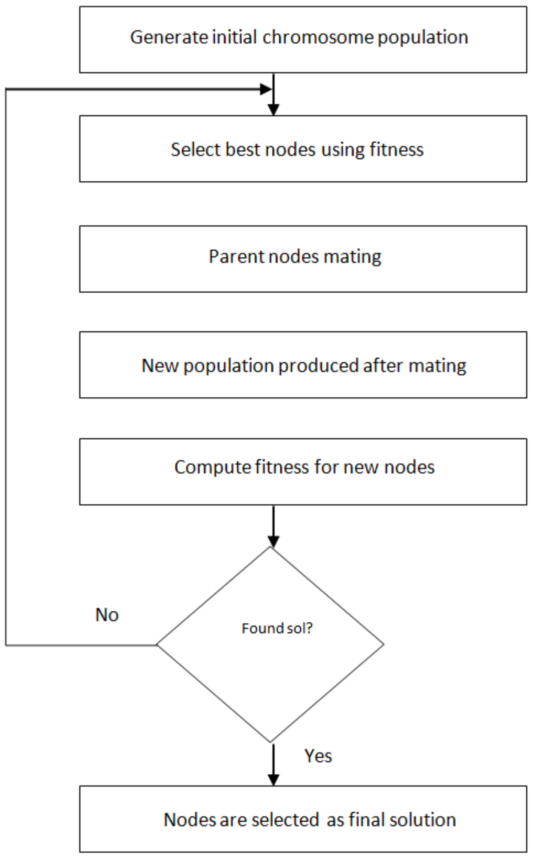
Generate initial chromosome population
Select best nodes using fitness
Parent nodes mating
New population produced after mating
Compute fitness for new nodes
Found sol?
No
Yes
Nodes are selected as final solution

Fig.1: Modified Genetic Algorithm – Block Diagram

Selection

The mechanism selects a solution for moving ahead in creating next generation nodes. This is a key operation to increase the productivity of the nodes. Several methods are used to select accurate nodes several selection strategies (Goyal & Agrawal 2013).

Crossover

Crossover operation creates individual nodes by reforming their parent nodes using hybridization method.

Mutation

Mutation is the next step that introduces diversity in the groups. This is applied to homogeneous population. This process alters the gene values of the nodes from its initial state. This results in a new set of values being added to the resource pool. With this the MGA may be able to get a optimal result.

The steps involved in Modified Genetic Algorithm for data replication is shown below,

- begin
- generate initial population with the selected nodes
- while (node_count > 0)
- calculate fitness F by calculating distance between two nodes
- compute F= inverse of min(Ed(m,n))
- evaluate each of the node for fitness
- do
- i) select parent node
- ii) crossover of parent node
- iii) mutation of descendant nodes
- iv) inspection of new nodes
- v) select new nodes
- Goto to step 2 until best solution obtained
- end

The problem of data replication without considering their size and capacity may result in poor solution. The MGA algorithm is able to produce an improved replication cost than earlier genetic algorithm by improving the fitness function which is able to handle dynamic data and frequent data updates.

Greedy Search Algorithm (GSA)

Greedy algorithm selects the best option in each iteration so as to proceed with best results which improves the performance. This GSA method divides a large problem into smaller and smaller problems solution will get a global solution (Pan *et al*., 2016). Greedy algorithm uses the minimum spanning tree in the chart and Huffman encoding. It can be used as alternative where precise result is not required

The methodology of greedy algorithm is: Begins with a primary solution of a problem to approach final goal in a step by step manner. It will stop at a point called saturation level.

A simple method of data replication using greedy search technique is created. For each number of site Zi and item Uk, the repetition function value Ak is defined as

$$A_k^i = \frac{R_k^i - \left(\sum_{x=1}^{m} W_k^x U_k C(i,Z) - W_k^i\right)}{U_k}$$

The above repetition function Ak represents the replication benefit in terms of network cost, if the item Uk replicated at Zi. This benefit is that the value Ak is computed by using the difference between the communication cost occurred from the current read requests. So it reduces the communication cost by half than the previous methods. However, for the adverse value of replication function Ak replicating the ith site in its local view of kth site will not produce an accurate result. This does not mean that network cost will be be always at its peak. But improving the site parameters the network cost can be managed.

A list Li is created for Ui containing all the nodes that are replicated. A node Nk can be replicated only when the storage capacity bi of the site is greater than the replication space requested by the node. A positive value should be generated while finding the difference betwwen the storage space and requsted replication space. Additionally, a list AL is maintained to include any additional resources provisioned in cloud and thus the replication capacity is improved.

The steps involved in Greedy Search algorithm is shown below,

- begin
- create list Li and additional list L1
- while (Li and L1 ≠ NULL)
- Initialize Um=0, Nm={}
- /*Um contains present max value of Ak,
- Nm preserves node identity */
- Select a node Uk belongs to list L1 in circular order
- for each Nk in Li
- calculate replication function Ak
- if (Um<Aki) then Um= Aki and Nm=k
- else
- Li = Li – {Nk}
- /* performes node update with neighbours */

- for nodes in L1 change SNmi
- /* New nodes included */
- Ai= Ai - Nk
- /* Remove resource on replication completion */
- If Li = NULL then L1=L1- {Ui}
- end while

This Greedy search replication algorithm generates improved results but it cannot guarantee that the final solution is the best. The MGA algorithm outperforms this algorithm in terms of capability of handling large data volume. The greedy algorithm can be applied extensively but cannot guarantee a optimal solution unlike genetic algorithm.

Results and Discussion

In this section, Modified Genetic Algorithm (MGA) and Greedy Search Algorithm (GSA) is compared with the random method for the data replication performance. Python programming language is used for implementing these algorithms A cloud environment with a range of minimum of 250 nodes to a maximum of 1000 nodes is taken for the experimentation. Evaluation results have been computed in terms of time taken for worst case recovery, average case recovery, recovery of 20% corrupt data and recovery of 40% corrupt data as shown in tables 1 to 4 and figures 2 to 5.

Table 1: Time taken for recovering worst case data corruption

Number of nodes in Cloud	Random	MGA	Greedy Search
250	2542	2021	2502
500	3668	3301	3522
750	5209	4674	5108
1000	8142	6822	8022

Table 2: Time taken for recovering average case data corruption

Number of nodes in Cloud	Random Method	MGA	Greedy Search
250	14	11	13
500	17	14	16
750	21	19	20
1000	27	24	25

From the figure 2, it can be observed that the Modified GA has lower worst case recover time of 22.83% than the random method and 21.68% than GSA for 250 nodes, 10.53% lesser than the random method and 8.87% than GSA for 500 nodes, 10.82% lesser than the random method and 8.87% than GSA for 750 nodes and 17.64% lesser than the random method and 16.16% than GSA for 1000 nodes.

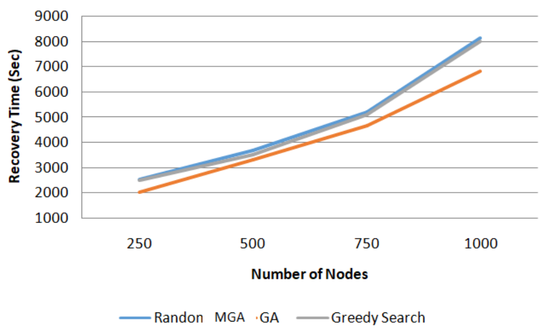

Fig. 2: Time taken for recovering worst case data corruption

From the figure 3, it can be observed that the MGA has lower average case recover time of 24% than the random method and 16.66% than GSA for 250 nodes, 19.35% lesser than the random method and 13.33% than GSA for 500 nodes, 10% lesser than the random method and 5.12% than GSA for 750 nodes and 11.76% lesser than the random method and 4.08% than GSA for 1000 nodes.

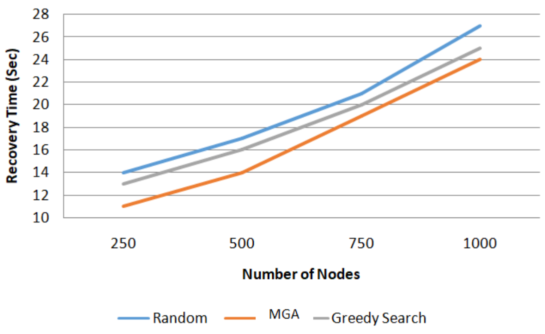

Fig. 3: Time taken for recovering worst case data corruption

From the figure 4, it can be observed that after 20% of data corruption MGA has lower recovery time by 18.18% than the random method and GSA for 250 nodes, 14% lesser than the random method and 13.33% than GSA for 500 nodes, 31.57% lesser than the random method and 22.22% than GSA for 750 nodes, by 20% lesser recovery time than random method and GSA for 1000 nodes.

Table 4: Time taken for recovering after 40% of data corruption

Number of nodes in Cloud	Random Method	MGA	Greedy Search
250	7	7	6
500	9	7	8
750	13	9	11
1000	13	12	11

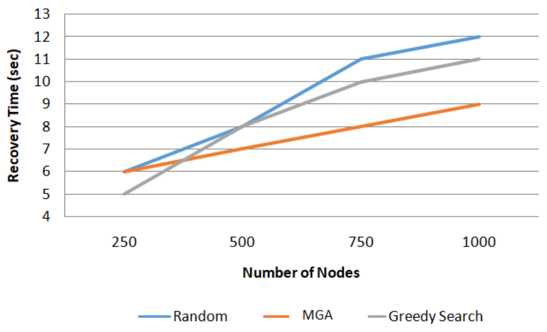

Fig. 4: Time taken for recovering after 20% of data corruption

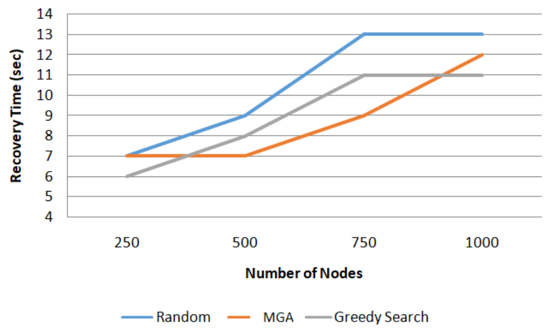

Fig. 5: Time taken for recovering after 40% of data corruption

From the figure 5, it can be observed that after 40% of data corruption MGA has same recovery time as that of random method and 15.38% higher than the GSA for 250 nodes, 25% lesser than the random method and 13.33% than GSA for 500 nodes, 36.36% lesser than the random method and 20% than GSA for 750 nodes and 8% lesser than the random method and 8.69% than GSA for 1000 nodes.

Conclusion

In this work, we propose the Modified Genetic Algorithm (MGA) and Greedy SearchMethod (GSM) to handle the drawbacks existing in the previous schemes. Both of these algorithms are based on deterministic approach in whichlag in accessing remote data is reduced by storing the data closer to the services. The GSA approach considers an initial solution for a problem and fixesa set of goal in a step by step manner so as to obtain better solutions in a least time.The MGA method chooses a particular set of data blocks and applies the fitnessfunction which selects best nodes for replication. Both of these methods find outthe best nodes for replication, but they select a particular set of data blocks. If thebest nodes are not available in those blocks, they will choose next set of blockswhich results in increase of replication cost. But dynamic data are effectivelyhandled by these methods by using genetic approach for population selection.

Acknowledgement

We acknowledge the support provided by The Professor and Head, Dept. of Physical Sciences and IT, TNAU, Coimbatore.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

Conflict of Interest

The authors do not have any conflict of interest.

References

1. Boru, D., Kliazovich, D., Granelli, F., Bouvry, P., & Zomaya, A. Y. (2015). Energy-efficient data replication in cloud computing datacenters. *Cluster computing,* 18(1), 385-402.

2. Liu, C. W., Hsien, W. F., Yang, C. C., & Hwang, M. S. (2016). A Survey of Public Auditing for Shared Data Storage with User Revocation in Cloud Computing. IJ Network Security, 18(4), 650-666.
3. Zhang, J., & Dong, Q. (2016). Efficient ID-based public auditing for the outsourced data in cloud storage. *Information Sciences,* 343, 1-14.
4. Wang, B., Li, B., Li, H., & Li, F. (2013, October). Certificateless public auditing for data integrity in the cloud. In Communications and Network Security (CNS), 2013 IEEE Conference on (pp. 136-144). IEEE.
5. Lin, J. W., Chen, C. H., & Chang, J. M. (2013). QoS-aware data replication for data-intensive applications in cloud computing systems. IEEE Transactions on Cloud Computing, 1(1), 101-115.
6. Boru, D., Kliazovich, D., Granelli, F., Bouvry, P., & Zomaya, A. Y. (2015). Energy-efficient data replication in cloud computing datacenters. Cluster computing, 18(1), 385-402.
7. Zhang, Y., Xu, C., Li, H., & Liang, X. (2016). Cryptographic public verification of data integrity for cloud storage systems. IEEE Cloud Computing, 3(5), 44-52.
8. Williams, H., & Bishop, M. (2014). Stochastic diffusion search: a comparison of swarm intelligence parameter estimation algorithms with ransac. Algorithms, 7(2), 206-228.
9. El-henawy, I. M., & Ismail, M. M. (2014). A Hybrid Swarm Intelligence Technique for Solving Integer Multi-objective Problems. *International Journal of Computer Applications*, 87(3).
10. Huo, Y., Zhuang, Y., Gu, J., Ni, S., & Xue, Y. (2015). Discrete gbest-guided artificial bee colony algorithm for cloud service composition. *Applied Intelligence*, 42(4), 661-678.
11. Połap, D., Woźniak, M., Napoli, C., & Tramontana, E. (2015). Real-time cloud-based game management system via cuckoo search algorithm*. International Journal of Electronics and Telecommunications*, 61(4), 333-338.
12. Florence, A. P., & Shanthi, V. (2014). A load balancing model using firefly algorithm in cloud computing. *Journal of Computer Science,* 10(7), 1156.
13. Jacob, L., Jeyakrishanan, V., & Sengottuvelan, P. (2014). Resource scheduling in cloud using bacterial foraging optimization algorithm*. International Journal of Computer Applications*, 92(1).
14. Hu, X. (2015). Adaptive optimization of cloud security resource dispatching SFLA algorithm. Int. J. Eng. Sci.(IJES), 4(3), 39-43.
15. Dashti, S. E., & Rahmani, A. M. (2016). Dynamic VMs placement for energy efficiency by PSO in cloud computing. *Journal of Experimental & Theoretical Artificial Intelligence,* 28(1-2), 97-112.
16. Zhou, J., & Yao, X. (2017). A hybrid approach combining modified artificial bee colony and cuckoo search algorithms for multi-objective cloud manufacturing service composition. *International Journal of Production Research,* 55(16), 4765-4784.
17. Zhou, J., & Yao, X. (2017). A hybrid artificial bee colony algorithm for optimal selection of QoS-based cloud manufacturing service composition. *The International Journal of Advanced Manufacturing Technology*, 88(9-12), 3371-3387.
18. Cavazos, J., Moss, J. E. B., & O'Boyle, M. F. (2006, January). Hybrid optimizations: Which optimization algorithm to use?. In Compiler Construction (pp. 124-138). Springer Berlin Heidelberg.
19. Sedano, A., Sancibrian, R., de Juan, A., Viadero, F., & Egana, F. (2012). Hybrid optimization approach for the design of mechanisms using a new error estimator. *Mathematical Problems in Engineering*, 2012.
20. Khanam, R., Kumar, R. R., & Kumar, C. (2018, March). QoS based cloud service composition with optimal set of services using PSO. In 2018 4th International Conference on Recent Advances in Information Technology (RAIT) (pp. 1-6). IEEE.
21. Dai, Y., Lou, Y., & Lu, X. (2015, August). A task scheduling algorithm based on genetic algorithm and ant colony optimization algorithm with multi-QoS constraints in cloud computing. In Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2015 7th International Conference on (Vol. 2, pp. 428-431). IEEE.
22. Arindam Das and Ajanta De Sarkar, "On Fault Tolerance of Resources in Computational

Grids", *International Journal of Grid Computing and Applications*, Vol.3, No.3, Sept. 2012, DOI: 10.5121/ijgca.2012.3301.

23. Grover J., Sharma, M.: Cloud computing and its security issues—a review (2014).
24. Kulkarni, G., Waghmare, R., Palwe, R., Waykule, V., Bankar, H., &Koli, K. (2012, October). Cloud storage architecture. In Telecommunication Systems, Services, and Applications (TSSA), 2012 7th International Conference on (pp. 76-81). IEEE.
25. Modi, C., Patel, D., Borisaniya, B., Patel, A., Rajarajan, M.: A survey on security issues and solutions at different layers of Cloud computing, pp. 1–32 (2012).
26. Mukundha, C., &Vidyamadhuri, K. (2017). Cloud Computing Models: A Survey. Advances in Computational Sciences and Technology, 10(5), 747-761.
27. Nepal, Surya, *et al*. "DIaaS: Data integrity as a service in the cloud." Cloud Computing (CLOUD), 2011 IEEE International Conference on. IEEE, 2011.
28. NIST Definition of Cloud Computing- https://www.nist.gov/programsprojects/cloud-computing