International Journal of Engineering & Technology, 7 (4) (2018) 2687-2691

SPC

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i4.14135
Research paper

Cloud hosting services and resources utilizing efficient proxy server based speculative algorithm

Ravi Mahadevan [1] *, Neelamegam Anbazhagan [2]

[1] *Research Scholar, Dept. of Computer Science and Engineering, Alagappa University, Karaikudi, Tamilnadu, India*
[2] *Professor , Alagappa University, Karaikudi, Tamilnadu, India*
**Corresponding author E-mail: ravimahadevan.phd@gmail.com*

Abstract

Cloud computing is an envisioned technique for service providing paradigm. Currently, many enterprises and organizations are hosting data into cloud service provider to minimize the maintenance cost, energy consumption and improve the data consistency. However, the several cloud servers facing the various hosting service policies, clients may be confused with which cloud server providers are appropriate for storing data and which hosting methodology is inexpensive. The current status quo is clients frequently put data into the single cloud and simply trust to luck. In the paper, proposes a new data hosting redundancy methodology called as Efficient Proxy Server Based Speculative (EPSBS) algorithm that provides accessibility guarantee, reducing hosting service, energy consumption and enhances Quality of Services (QoS) performance concurrently and it requires integrating two key functions. The first function is choosing numerous appropriate cloud service providers and a suitable EPSBS redundancy methodology to store data with guaranteed accessibility. The second key function is triggering a transmission process to reallocate data along with the deviations of data access prototype of clouds. Based on Experimental evaluations, proposed methodology reduces 0.11 ET (Execution Time), and 0.02 EC (Energy Consumption) and improves 29% HS (Hosting Service size), 2.8 T (Throughput) compared than existing methodologies.

Keywords:*Cloud Computing; Efficient Proxy Server Based Speculative (EPSBS) Algorithm; Quality of Services (QOS); Cloud Service Provider; Execution Time; Energy Consumption; Hosting Service Size; Throughput*.

1. Introduction

Cloud services recommend web hosting to individual or organizations based on their requirements through pay-as-you-go service as well as free hosting services with limited features. Hosting services consist of two: Smaller hosting services and larger hosting services. Mostly larger hosting services are coordinated for bigger organizations to authentically store data on web servers like cloud environment to access whenever necessary including technical support whereas smaller hosting services provide limited usage space. The host management process follows using web-rack servers allocated for every data entry in cloud storage simultaneously and increases performance factors through response time and bandwidth utilization. Cloud provides efficient storage for every consumer by ensuring data reliability, scalability, and budgeted hosting services.

Current pricing and scheduling methods does not consider price-to-performance assurance to the clients. It is unable to capture the inherent conflict of interests among different clients. Some faults occur during resource allocation. In this, some jobs are not allocated to virtual machines; some jobs are allocated to multiple virtual machines. It is chance to waste the resource and time. The existing system lacks in processing time, bandwidth utilization, energy consumption and resource allocation, from various web servers, which is not comfortable for consumers. Most cloud service providers utilized the cloud for hosting service, and it cannot handle multiple consumers' to access/upload data to the cloud at the same instance.

To overcome existing issues that propose a new data are hosting redundancy methodology called as Efficient Proxy Server Based Speculative (EPSBS) algorithm that provides accessibility guarantee, reducing energy consumption and enhances hosting service size, Quality of Services (QoS) performance concurrently and it requires integrating two key functions. The first function is choosing numerous appropriate cloud service providers and a suitable EPSBS redundancy methodology to store data with guaranteed accessibility. The second key function is triggering a transmission process to reallocate data along with the deviations of data access prototype of clouds. The paper contributions are following as:

- To develop an EPSBS algorithm that can provide accessibility guarantee improves hosting services size concurrently.
- To select numerous appropriate cloud service providers and a suitable redundancy methodology for storing data with guaranteed accessibility.
- To trigger a transmission process to reallocate data along with the deviations of data access prototype of clouds.
- To maintain resources for utilizing expected load and a buffer for typical load deviations.
- To reduce Execution Time, energy consumption and improves hosting service size and throughput compare than existing approaches.

The rest of paper contribution is followed by section 2 explains the literature study which is the closest to the proposed system. Section 3 discusses proposed design, proposed work and algorithms details. Section 4 discusses proposed experimental setup, evaluation parameters and comparative analysis with existing

approach. Section 5 concludes overall work with the future outcome.

2. Related work

Zhenhua Li et al. [1] implemented an integration of allocation algorithm and fast-convergent iterative algorithm which increased bandwidth efficiency and performance of the system to download files simultaneously utilized big data technology. So, the paid vendor was only able to utilize service with technical support and difficult to fetch un-popular files. Maxim et al. [2] described a cloud infrastructure service mechanism, and it efficiently balanced the load through other servers concurrently, and availability of data was necessary for consumers to utilize pay-per-use service. The service utilized appropriate algorithm for comparative analysis which improved data reliability and performance.

Mazhar Ali et al. [3] developed a data security approach for storage where un-authorized access was revoked utilized third-party key management scheme that generates a secret key for accessing files securely in a cloud environment. However, the third party was knowledge about key management scheme which reveals data information. Zhenhua Li et al. [4] discussed the cloud synchronization mechanism and storage utilization among various cloud providers efficiently. It initialized services to synchronize data of mobile devices into cloud storage like Google drive, dropbox and so on. Since the traffic congestion is enormously high while synchronizing data.

Karthikeyan et al. [5] analyzed hybrid cloud infrastructure and reduced migration and pricing cost through resource allocation of every server depends on the size of data stored and resource utilized. Although, the hybrid cloud only reduced cost and it does not reduce the energy consumption and bandwidth utilization. Christopher Hansen et al. [6] designed a cloud-RAID architecture evaluated through distributed cloud infrastructure providers that increased data reliability, security and data availability in cloud storage. Still, the performance was not yet tested by utilizing implemented architecture in the cloud.

Aditya KaushalRanjan et al. [7] developed a file assured deletion (FADE) approach, and it increased security in the cloud where authenticates file access by secret key generation mechanism for every authorized consumer. Though the private key was provided by the key manager, the information easily modified. Hendrik et al. [8] implemented access control management (ACM) technique which ensured technical support and avoids cross-service providers. It improvised access control mechanisms that increased system performance via secure key generation with ACM. However, the API management was not completely designed for cloud services where it contained limited languages.

Awad et al. [9] designed a chaotic fuzzy transformation methodology, and it supported a protected fuzzy keyword indexing, storage and query. It is managing the privacy and confidentiality of the user data and storing the resources of the cloud users. However, the ranking was manually performed.

3. System methodology

The proposed design illustrates the EPSBS framework is data hosting methodology for various cloud infrastructures and reducing cost from the framework while selects the appropriate cloud service provider and proper redundancy methodology to store data. The main methodology objective is to hosting data into the cloud, minimizing the energy consumption, cost and minimizes resource utilization. The proposed methodology illustrates the system architecture with implementation steps and proposed algorithm details. Figure 1 expresses the architecture of implementation process in details. The method enhances the QoS of CSP in the aspects of a cloud resource.

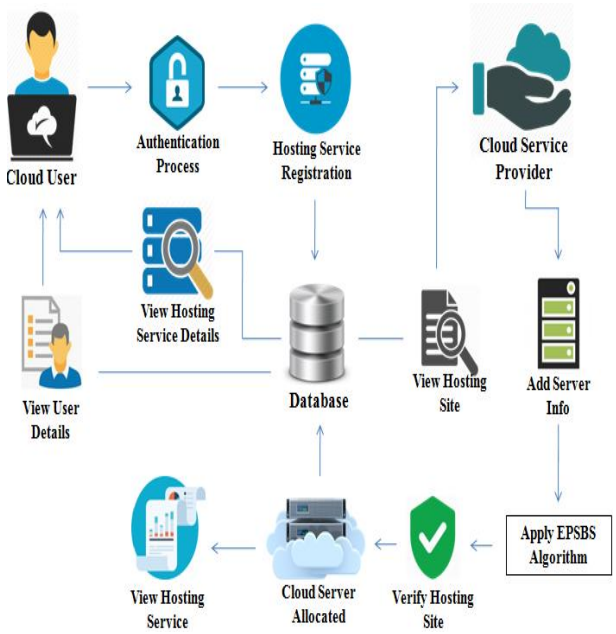

Fig.1: Architecture Diagram of EPSBS Algorithm.

3.1. Cloud service performance

Many data centers are allocated around the globe. Same or different kind of data centers includes similar or differentiated cloud providers. Consequently, every data center accessed by the client in a certain area, but the client should satisfy with the data in different performance. The response time of few data centers is very low and that the few ones may be insupportable is high. The EPSBS algorithm selects a particular cloud for particular data which is appropriate to the file and information which provides good throughput and overcomes the restriction of the current framework without affecting other data. After that, the cloud server has less workload of the task and delays the increase of operations, and it executes another data request. The storage mode transmission does not crash the performance of cloud services. To reduce the priority of transmission tasks and develop the transmission in the group when the proxy has less workload.

3.2. Hosting service

In the data, hosting model described with huge accessibility in various cloud storage services such as EPSBS mechanism. In the model is situated in the proxy. There are four major parts of EPSBS algorithm such as data hosting, storage manner switching, the statistic of workload, and analyst. The statistic of workload keeps gathering and tackling access logs to perform the placement of information. It additionally transmits statistic data to analyst which monitors the action of storage manner switching. The data hosting of data stores information utilizing replication or removal coding depending on the size and access occurrences of the data. The storage manner switching selects whether the storage manner of certain data should be modified from replication to removal coding or defeats explanation of outcome analyst. The hosting of data selects storage manner category and the cloud service providers and that the data should be stored. The analyst utilized to calculate the future access occurrences of data.

3.3. Security of cloud storage

It utilizes the accessibility declares in the SLAs of cloud storage services. The majority of the proposed methodologies imagine a third party or the cloud storage service providers managing key management is believed and cannot be hacked. Several authorities may force cloud storage providers to disclose client secrets or confidential information on the cloud storage, therefore altogether avoiding storage encryption strategies. A few entities may inter-

rupt communications among the clients and cloud storage providers and after that induce the cloud storage providers to release client secrets through utilizing illegitimate performances. The encrypted information is imagined and recognized, and the cloud storage providers are demanded to discharge client secrets. To circumvent such conditions, a particular type of encryption strategy utilized such as declinable encryption strategy. It utilized objectives at constructing an encryption strategy which could support cloud storage providers avoid this difficult situation. In the strategy, the cloud storage providers generate forgery client account and that the external authorities can simply acquire faked data from clients stored encryption data.

3.4. Server module

In the server, the module is to upload the data files for utilizing a few access policies. Initially, it obtains the public key for uploading specific data file; after receiving the public key, data owner demands the secret key for upload their specific data file.

3.5. Client or File download module

The module utilized to support the client to search data file utilizing the file id and file name. The client's can download the data file from the cloud storage server in anywhere and anytime. For downloading the data, the client should select the data file from the application; where the cloud server select the corresponding cloud service provider details from the connected database and also check the cloud server availability for downloading. If the file id and name are incorrect, then the data file is not downloading; otherwise the server asks the public key and get the encryption data file information. If the client needs the original data then they should have the secret key to get the original data file.

3.6. Efficient proxy server based speculative (EPSBS) algorithm

Every cloud value is assigned and is computed based on four aspects such as availability, storage, resource utilization and processing time and it specifies the preference of cloud storage. To select the majority of selected clouds and after that speculatively exchange the cloud storage in the preferred set with the cloud storage in the complementary set to search the best resolution.
The accessible cloud storage containing cost details and accessibility and the storage manner selected cloud services with the input data files size and read count. The proxy server obtains a data file from the cloud storage services anywhere the data file originally stored, and locates the data file into the newly selected cloud storage services utilizing new storage mode. The development of the algorithm computed outgoing bandwidth, energy, resource utilization and processing time of numerous processes. The storage mode computed in advance since it is influenced only through the access of cloud storage services, cost policies and accessibilities. While selecting the storage mode for every data file and utilize the real occurrence and the size of the data file. The pseudo code of proposed algorithm is explained below in details.
Input: Client and Cloud service provider
Output: Compute Bandwidth utilization, energy consumption, resource utilization and processing time.
Procedure:
Start
Cloud service provider (CSP) and client authentication process
Clients view plan of the resources
Service Hosting Registration
Cloud Service Provider
View hosting site
Add Server
Apply EPSBS Algorithm
If Verify Hosting Site
Hosting Service Created
View Hosting Service
Else
Hosting site is not created
End if
View list of hosting service;
Display Execution time, energy consumption, throughput and hosting service size
End

4. Result and discussion

4.1 Implementation setup

The proposed methodology is deployed with Intel i6 Core processor, with 16 GB RAM, 500 GB Memory with Windows7 Ultimate operating systems. The proposed algorithm is implemented in Java programming environment by using NetBeans 8.0, JDK (Java Development Kit) 1.8, MYSQL database 5.5 in Java programming environment.

4.2. Performance matrix

The proposed algorithm is calculated in different kinds of constraints to calculate the efficiency of techniques. The proposed technique is highly dedicated to providing efficient content data file storing with data accessibility in QoS guarantee services. The proposed algorithm is computed with following constraints namely Hosting Service, Execution Time, Throughput and Energy consumption.

4.2.1. Hosting service (HS)

The hosing service is described as the Difference between some requests processed and summation of utilizing size of data file and size of hosting plan size. The hosting service is computed in Equation (1).

$$HS = \frac{1}{no.ofrequest} \sum_{i=1}^{no.ofrequest} [\frac{sizeofHS - sizeofDF}{sizeofHS}] \quad (1)$$

Where DF is data file.

4.2.2. Throughput

Throughput is the difference between a total number of request processed and subtraction of end time and start time. The throughput is calculated in Equation (2).

$$Throughput = \frac{(Total\ no.ofrequestproceed)}{EndTime - StartTime} \quad (2)$$

4.2.3. Energy consumption (EC)

The energy consumption is summation of some request processed and the size of the data file and constant energy. The energy consumption is computed in Equation (3)

$$EC = \sum_{i=1}^{noofrequest} [sizeofDF * Energy] \quad (3)$$

Where DF is data file.

4.2.4. Execution time (ET)

The execution time is Difference between some requests processed and maximum execution time and summation of ET of hosting service size and size of data file. The Bandwidth Utilization is estimated in equation (5).

$$ET = \frac{1}{\max(ET) * no.ofrequest} \sum_{i=1}^{n.ofrequest} SizeofHS * SizeofDF \quad (5)$$

Where, HS is hosting service and DF is data file.

Table 1 explains the Hosting Service (HS), Energy Consumption (EC), Execution Time (ET) and Throughput (T) for respective input parameters with existing methods. Table 1 displays the average value on all respective evaluation matrix and input parameters with Genetic Algorithm (GA) [10], Particle Swarm Optimization (PSO) [10] and Intelligent Bee Colony Algorithm (IBC) [10] existing methodologies. According to Table 1, it noticed that proposed EPSBS Algorithm performs well on all evaluation matrix and Input parameters compare than existing methods.

Table 1: Average Delay, Energy Consumption, Execution Time, Bandwidth Utilization for Dynamic Routing

Algorithm	Hosting Service (%)	Execution Time (s)	Energy Consumption (s)	Throughput (Mbps)
GA	35	0.47	0.166	5.25
PSO	45	0.43	0.125	5.75
IBC	65	0.41	0.09	6.70
EPSBSA	94	0.30	0.07	9.5

Figure.2 demonstrates the Hosting Service, and the proposed algorithm EPSBS is comparing existing algorithms such as GA, PSO, and IBC. The storing of a data file from cloud storage service utilizing EPSBS algorithm produces less hosting service time when it compared to existing methodologies.

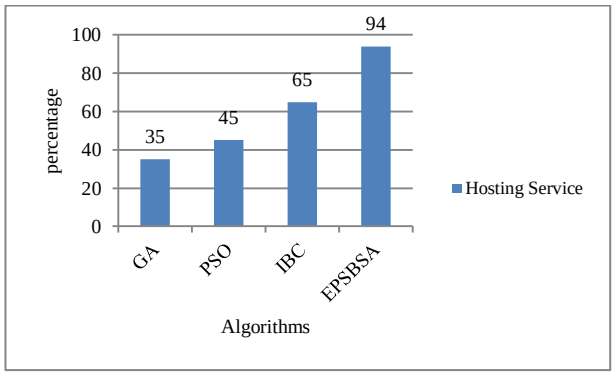

Fig. 2: Hosting Service of Cloud.

Figure.3 demonstrates the throughput in kbps, and the proposed algorithm EPSBS is comparing existing algorithms such as GA, PSO, and IBC. The storing of a data file from cloud storage service utilizing EPSBS algorithm produces less execution time when it compared to existing methodologies.

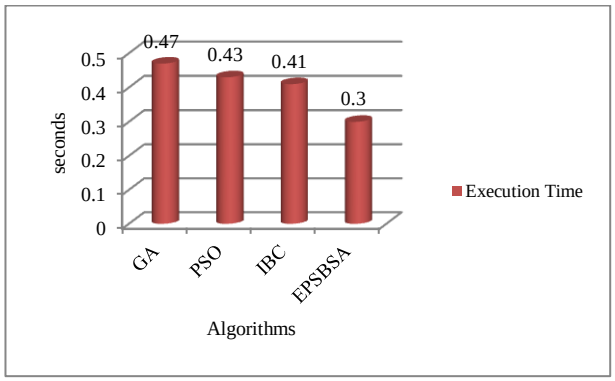

Fig. 3: Execution Time of Hosting Service.

Figure.4 demonstrates the energy consumption in joules, and the proposed algorithm EPSBS is comparing existing algorithms such as GA, PSO, and IBC. The storing of a data file from cloud storage service utilizing EPSBS algorithm consumed low energy when it compared to existing methodologies.

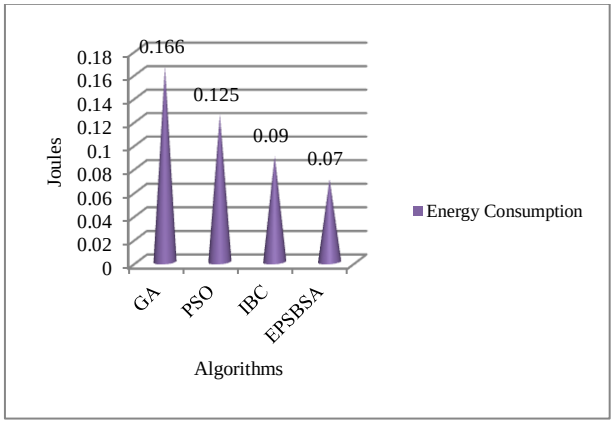

Fig. 4: Energy Consumption of Hosting Service.

Figure.5 demonstrates the throughput, and the proposed algorithm EPSBS is comparing existing algorithms such as GA, PSO, and IBC. The storing of a data file from cloud storage service utilizing EPSBS algorithm provided high throughput when it compared to existing methodologies

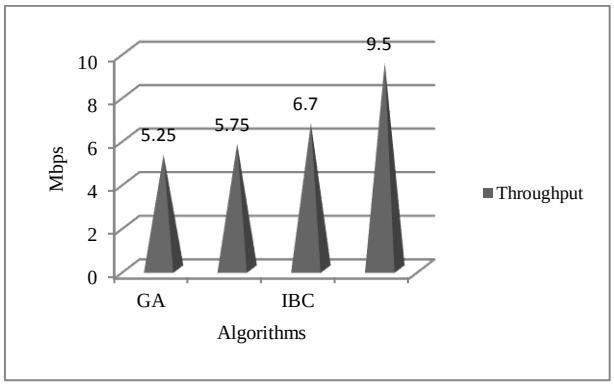

Fig. 5: Throughput of Hosting Service.

Based According to Figure [2] to [5] observations, the proposed methodology computes Hosting Service, execution time, energy consumption and throughput for finding the efficiency. Proposed EPSBSA is estimated with GA [10], PSO [10] and IBC [10] existing techniques behalf of Hosting Service, execution time, energy consumption and throughput. The proposed methodology is the closest competitor is IBC algorithm on overall parameters. IBC techniques improved resource sharing in cloud environments. It improved the performance of various IT resource environments to store data. However, IBC methodology failed to maintain service allocation information. Proposed method improves the throughput of hosting service. Proposed EPSBS reduces 0.11 ET (Execution Time), and 0.02 EC (Energy Consumption) and improves 29% HS (Hosting Service), 2.8 T (Throughput). Finally, the paper claims the proposed EPSBS algorithm performs best on every evaluation matrix and respective input parameters.

5. Conclusion

The EPSBS algorithm is guaranteed high accessibility, a novel cloud storage strategy that supports clients to allocate data between cloud service providers effectively in huge, physically allocated and highly dynamic infrastructures. The request of transference and delocalization of data is developing, the cloud storage services are practicing the quick improvement and the cloud storage services based on the cloud. A main challenge of the paper is moving cloud storage services to the cloud service providers are the hosting service participation. The EPSBS algorithm generates the best resolution to the difficulties relating to which cloud storage manner to utilize and which cloud storage providers to put the data. It demands the specific cloud infrastructure and hosting the

data file into clouds though guaranteeing flexible accessibility and circumventing service retailers lock-in. Proposed EPSBS algorithm reduces 0.11 ET (Execution Time), and 0.02 EC (Energy Consumption) and improves 29% HS (Hosting Service), 2.8 T (Throughput). Finally, the paper claims the proposed EPSBS methodology performs best on every evaluation matrix and respective input parameters.

References

[1] Li, Z. H., Liu, G., Ji, Z. Y., & Zimmermann, R., "Towards cost-effective cloud downloading with tencent big data", *Journal of Computer Science and Technology*, Vol. 30, No. 6, pp. 1163-1174, 2015.https://doi.org/10.1007/s11390-015-1591-5.

[2] Schnjakin, M., Metzke, T., &Meinel, C., "Applying erasure codes for fault tolerance in cloud-raid", *In Computational Science and Engineering (CSE), 2013 IEEE 16th International Conference on IEEE*, pp. 66-75, 2013.

[3] Ali, M., Malik, S., & Khan, S., "DaSCE: Data security for cloud environment with semi-trusted third party", *IEEE Transactions on Cloud Computing*, 2015.

[4] Li, Z., Zhang, Z. L., & Dai, Y., "Coarse-grained cloud synchronization mechanism design may lead to severe traffic overuse", *Tsinghua Science and Technology*, Vol. 18, No. 3, pp. 286-297, 2013.https://doi.org/10.1109/TST.2013.6522587.

[5] Karthikeyan, T., &Nandhini, T., "Dependent component cost model of legacy application for hybrid cloud", *In Circuit, Power and Computing Technologies (ICCPCT), 2016 International Conference on IEEE*, pp. 1-5, 2016.

[6] Hansen, C., & Archibald, J., "The Case for RAID 4: Cloud-RAID Integration with Local Storage", *In Availability, Reliability and Security (ARES), 2016 11th International Conference on IEEE*, pp. 235-240, 2016.

[7] Ranjan, A. K., Kumar, V., & Hussain, M., "Security Analysis of Cloud Storage with Access Control and File Assured Deletion (FADE)", *In Advances in Computing and Communication Engineering (ICACCE), 2015 Second International Conference on IEEE*, pp. 453-458, 2015.

[8] Graupner, H., Torkura, K., Berger, P., Meinel, C., &Schnjakin, M., "Secure access control for multi-cloud resources", *In Local Computer Networks Conference Workshops (LCN Workshops), 2015 IEEE 40th IEEE*, pp. 722-729, 2015.

[9] Awad, A., Matthews, A., Qiao, Y., & Lee, B., "Chaotic Searchable Encryption for Mobile Cloud Storage", *IEEE Transactions on Cloud Computing*, 2015.

[10] LD, D. B., & Krishna, P. V., "Honeybee behavior inspired load balancing of tasks in cloud computing environments", *Applied Soft Computing*, Vol. 13, No. 5, pp. 2292-2303, 2013.https://doi.org/10.1016/j.asoc.2013.01.025.