International Journal of Engineering & Technology, 7 (4) (2018) 3306-3311

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i4.18024
Research paper



Broker decision verification system using MR ^{cloud} tree

Anupriya Koneru [1] *, Sreelatha M [2]

[1] Research Scholar, Department of Computer Science and Engineering, Acharya Nagarjuna University, Guntur
[2] RVR & JC College of Engineering, Guntur
**Corresponding author E-mail: smartykoneru@gmail.com[1]*
lathamoturi@rediffmail.com[2]

Abstract

In the present industry, there is a gap between the requirements of the Cloud Service Requester and the services offered by Cloud Service Providers. To bridge this gap, Cloud Service Brokers extended their service towards the selection of Cloud Service Provider for Cloud Service Requester. In this process, the Broker may deceit and selects a bribery Provider that completely affects the Requesters business. This assumption shows the necessity of checking the correctness of the Cloud Service Broker. This paper focuses on proposing a Broker Decision Verification System which depends on MR^{cloud} Tree. It works on multidimensional data to provide the cloud service requester with a facility to check the correctness of the Cloud Service Broker and also offers only a single provider to the Requester which could be done by Reputation Factor Value of the Provider. The performance of Broker Decision Verification System is compared with MB^{cloud} Tree.

Keywords: *Broker Deceit; Broker Decision; Merkle Tree; MR^{Cloud} Tree; Verification.*

1. Introduction

NIST [1] defines "Cloud broker as an entity that manages the use, performance, and delivery of cloud services and negotiates relationships between cloud providers and cloud consumers." Cloud Service Broker (CSB) can provide the services in three ways: Service Intermediation, Service Aggregation, and Service Arbitrage. By using the Cloud Service Broker, the organizations can relax while choosing a better Cloud Service Provider (CSP) for their requirements. A number of providers are accessible in the market. It will be difficult for the Cloud Service Requester (CSR) or Organizations to choose a better provider. So, the Cloud Service Broker helps the Cloud Service Requester to choose a better provider.

CSB is a trusted third party, which can analyze CSRs needs and can help in the process of choosing the CSP. CSB can choose the CSP which can offer services at a reasonable Price. CSB takes care of all billing aspects of the service. It can take care of SLA violations of CSPs. But it should be continued up to date in offering services, options etc. It is a mediator between the CSR and CSP which really increases a layer of complexity, even though the advantages of CSB really attract the present IT industry.

Multi-cloud is related to cloud computing only, but there is confusion about where it fits within the terminology of Private Cloud, Public Cloud, Hybrid Cloud and Community Cloud. The Multi-cloud has provided an environment, which really gives flexibility to the CSRs to choose different price values. The advantages of Multi-cloud attracted a wide range of IT industry. Using this technology, CSRs can work as usual during disasters; can have high availability, scalability, and much more cost benefits. An overview led by IDC [2] found that 86% of endeavors anticipate the requirement of a Multi-cloud approach within the next two years to support their solutions. Right scale conducted a survey by asking 930 IT professionals about their adoption of cloud technologies. As per this survey, 82% of organizations refer to use Multi-cloud. Multi-cloud differs from Hybridcloud by allowing organizations to choose multiple clouds for different services whereas, in Hybridcloud, it is like developing a solution that uses more than one cloud to perform a particular task which accesses both. In the Multi-cloud environment, CSBs role is to understand the needs of CSR, select the best suitable CSP, and establish communication between CSR and CSP. The topmost CSBs in the market are Appirio, AWS marketplace, Blue Wolf, Cloud compare, Cloud More, Cloud nation, Cloud Italia, Cloud Sherpas, Comcast Upware, Compatible One etc. According to Markets and Markets, the overall cloud brokerage market [3] can grow from $1.57 billion in 2013 to $10.5 billion by 2018.

Trust is a qualitative dimension, tightly related to the reputation of cloud service providers and users direct experience (CSRs). There is a need for trust in CSB because the CSRs are not interested to waste their time for selecting a specific CSP and negotiate price and all. So, they need to contact CSB for their services to select proper CSP on behalf of it. So, CSB should be a trusted third party. There are some providers, who can offer trust as a service to their clients. So, it is also treated as an essential service. But the present concern is trust in CSB, not as a service. CSB is a marketplace, which offers the necessary services to the CSRs and business to the CSPs to create gains from the trade. In this context, CSB brings CSRs and CSPs together but doesn't itself provide any service. CSP understands that their current good actions will be rewarded by future business and current bad actions will be penalized by lack of future business, then CSP will have an incentive to act in good faith. With this knowledge, the CSB is providing business to the CSP. The reputation of the CSP becomes an important incentive mechanism that provides trust on the CSPs. In the Multi-cloud environment, there are multiple Cloud Service providers to satisfy the needs of a CSR. Then here the CSB has to

Copyright © 2018 Anupriya Koneru, Sreelatha M. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

choose the appropriate CSP for CSR. This selection can be done based on the CSPs previous actions. So, Good reputed CSP has to be selected by CSB to serve the CSR. Practically the CSB has adopted some form of a reputation calculation system to select the CSP suitable for CSR's Request. But how a CSR trust the CSB decision in the selection of CSP? It motivates to design a verification mechanism for CSR to check the CSB's selection decision.

In the present study an innovative method for the verification of CSB's decision by allowing the CSR to verify Authenticity, Satisfiability, Integrity, and Completeness is proposed.

The rest of this is paper ordered as follows. Section 2 examines the associated work, Section 3 states the Problem statement, Section 4 gives an Overview on the Broker Decision Verification System (BDVS), followed by verification mechanisms in Section 5, and Section 6 bangs the performance analysis and results. In the final section, the conclusions and scope for future work is specified.

2. Related work

This work is related to verification of the trustworthiness of the CSB. In 2.1 the existing approaches for the verification of CSB is discussed and in 2.2 review different single provider selection mechanisms which need to be studied to select a single provider among multiple Providers in the Broker environment is discussed.

2.1. Approaches for verification of CSB

In the distributed environment there are number of approaches which are helpful in verifying the decision.

The signature-based methods are truly effective for verification but in performance aspect the hash-based verification techniques are found to be effective.

Jingwei Li et.al [4] proposed a verifiable cloud service selection mechanism which consists of CSB, Clients, CSP and Collector. In this work, Author stated that the collector collects the CSP profile and constructs the MB cloud Tree (Merkle B Tree) and MMB cloud Tree (Multicluster Merkle B Tree). Later the collector signs on the root of these trees and publishes it. Now collector distributes the profile Database along with tree to the CSBs. In that situation, every Broker having might be the same Database which is authenticated by the collector, which makes the CSBs business stationary.

Johannes Harungguan Sianipar and Christoph Meinel [5] proposed a technique for the verification of Broker. In this work, the fourth party is named as Verifier whose role is to verify the clustering result of CSPs done by the Broker. Verifier verifies all the properties of CSPs which are members of clusters by using Multi-agent system. Initially, before receiving any request from the Client, the Broker clusters the CSPs based on their properties, makes a hash of it and sends it to the Verifier. Verifier collects the properties of every CSP by using multi-agent systems which are running on the CSPs. After collecting the profiles it applies to hash and compares these values with the received hash values. If there is any deference identified then it sends back the suggestion to the CSB to change the profile of the CSP. After updating that, the Verifier would sign on it. In this, management and control of multi-agent systems which are running on CSPs is very difficult. Another difficulty is that the CSB suggests a group of CSPs as a result to the client request. Then the Client needs some information about the CSP like previous feedback which is ignored in this work.

2.2. Review on single provider selection mechanism

In the multi-cloud environment, most of the Brokers return a set of Providers who could satisfy the requirements of the Requester [6]. Practically it would be difficult for the Requester to choose one out of the suggested list of providers. To solve this problem, [7] suggested the RFV algorithm to choose a single provider by considering the feedback of the Provider. If the feedback given by the Requester is not trust worthy that affects the business of the Cloud Service Provider. So, in [8] author Proposed an algorithm called DTRFV which is used to evaluate the given feedback based on the performance.

3. Problem statement

In the Multi-cloud environment, Broker architecture consists of three entities. Cloud Service Requester (CSR), Cloud Service Broker (CSB), and Cloud Service provider (CSP). The process in this environment is stated below:

CSPs and CSRs register with the CSB which in turn assigns a local Id to each CSP and CSR to avoid Sybil attacks. CSB maintains the profiles of all these CSPs in a separate data table in its local database. Whenever CSR sends a request for CSP by clearly specifying all its requirements then CSB selects the set of providers who offer these services and selects the one whose Reputation Factor Value (RFV) is high. After the completion of taking the service, the CSR submits the feedback form to the CSB for CSP. CSB analyses the feedback form and generates a recommendation value based on the RFV algorithm. For the next request from the CSR, the CSB considers the past 'n' recommendation values and calculates the average of it to generate the Reputation Factor Value (RFV). This value would decide further business of that CSP. The problem is with the trustworthiness of the CSB.

In this process, the Broker may deceit in four ways:

- The broker may change the properties of the CSPs
- The broker may select some bribery CSP which cannot satisfy the request.
- The broker may change the recommendation values after receiving the request.
- The broker may not consider all the eligible CSPs.

These kinds of actions by CSB make the system vulnerable and affect the CSRs business, because the bribery CSP may mislead the CSR. So, CSR needs to have an option to check the correctness of the CSB. As per [4], check Authenticity, Satisfiability, and Completeness are needed. Along with these, checking even the Integrity is the prerequisite.

4. overview of broker decision verification system (BDVS) scheme

To check the correctness of the Broker (CSB), the CSR should check Authenticity of the CSPs profile and also allow the requester to check the signature of CSP. But the above-stated structure, the actual ID of the CSP is not known to the CSR. So, CSR can't identify the CSP and can't check the signature. Satisfiability is another parameter which the Cloud Service Requester needs to check. CSR needs to check whether the suggested CSP satisfies all the requirements specified in the Query. Completeness of the query would be checked, whether all the CSPs which could satisfy the request would participate in the selection process. The Cloud Service Requester need to confirm it that the suggested CSP is the one which is selected among all qualifying CSPs and not the bribery CSP. The Integrity of CSP needs to be verified by the requester to avoid changes in RFV or Recommendation value of CSP. All the above-stated facilities should be provided to the Cloud Service Requester. But the CSR can't spend that much of time always to check all these essential issues. So, the BDVS scheme shown in Figure 1 introduces an entity in the Architecture. That entity is named as Authority. It gives permission to the CSBs to start their business by providing trusted Certificate. It also has the right to access the Database of the CSB as a governing authority, when there is any request from CSR to check the correctness of the CSB. The Authority is a trusted entity whose role is to check all the above stated four parameters. In the BDVS Scheme, there are three phases.

Phase 1: Structure a Database

Phase 2: Query Process

Phase 3: Decision Verification
Structure a Database - Cloud Service Broker collects complete profiles of the CSPs who are registered and also the signature of the CSP on every profile. The CSB assigns a local ID and stores the details in the name of assigned ID. Every CSP stored in the local Database of CSB with ID, list of properties, signed properties, and CSPs Public Key. On the other hand, CSRs also register with CSP and it assigns a local ID to it.

Query Process - In this phase, CSB receives the request query from the cloud service requester which has already registered. Broker initiates the service selection algorithm and selects the best Cloud Service Provider. It sends the local ID of CSP, list of Properties and RFV to the Cloud Service Requester.
Decision Verification - During this phase, the Cloud Service Requester sends the request to the Authority to check the correctness of the Cloud Service Broker. Authority initiates the Verification algorithm and sends back the result to the Requester. Then the Requester will decide whether the CSB is trustworthy or not.

5. The scheme for MR cloud tree

The MR ^{cloud} Tree is a hybrid approach which is the combination of a Merkle tree and Rtree.

Merkle tree is a Hash tree [9] which is framed to build the Tree of hashed data values. In general, hashing is used for fast retrieval of data from the databases using shortened keys. It is also applied for encryption and decryption of digital signatures. The message digest which is a hashed value of the signature is generated.

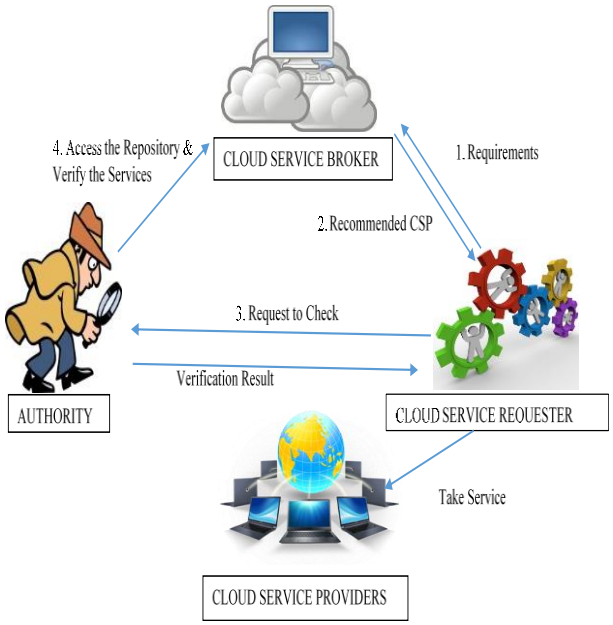


Fig. 1: Architecture of Broker Decision Verification System (BDVS).

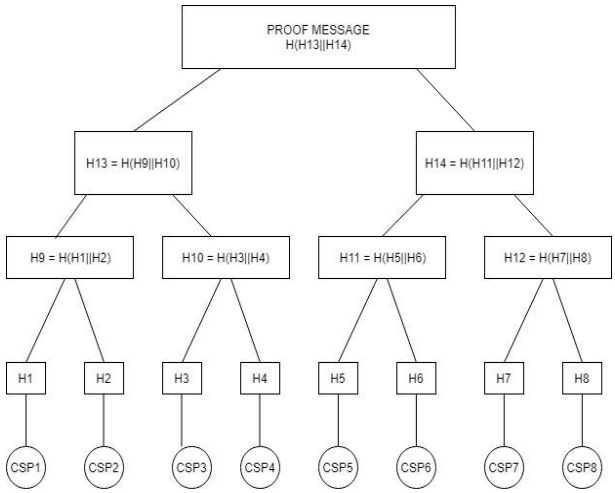


Fig. 2: Example for Merkle Tree.

Table 1: Sample 2D Data

Provider ID	Price	RFV	Provider ID	Price	RFV
CSP1	50	4	CSP7	10	4
CSP2	35	6	CSP8	30	2
CSP3	40	6	CSP9	20	3
CSP4	85	7	CSP10	45	5
CSP5	70	10	CSP11	10	1
CSP6	50	8	CSP12	100	8

The message digest and the signature are transferred over the network by the sender. On the other end, the receiver applies the same hash function to the received signature and generates the message digest. To verify the correctness, the received message digest is compared with the generated one at the receiving end. In this way, the hashing helps in cryptography. In this application, providing integrity for the profiles of cloud service providers and also to the RFV values is needed. For this purpose, a hash tree which is called a Merkle tree is used. In Figure 2 the details of eight cloud service providers have been taken and the Merkle tree has been constructed. The Root node of the tree consists of the Hash of all its decedents. It is called a proof message. It would be used in further verification.
The purpose of the R tree is to build an indexing structure for the fast retrieval of data from the database. R tree [10] is a dynamic index structure which is mostly used for spatial data. But the purpose of using that spatial data structure in our application is to handle multidimensionality.
In our application, multidimensional data is to be stored. So, it is not achieved by B trees which are restricted to one dimension. But B Trees provide a way to Region Trees (R Trees). To use R Trees the data is represented by a Minimum Bounding Region (MBR). R Trees can handle range queries very efficiently. In the present work, the Cloud Service Requesters specify their requirements in the form of range queries.
To understand the proposed methodology, 2D data which is a point in the spatial structure is considered. The sample data is given in Table1.

5.1. Structure a database

For two dimensional data, the R Tree with MBR constructed is shown in Figure 3. In this, every point represents a Cloud Service Provider profile in 2 Dimensional space, where the X-axis represents Price and Y-axis shows RFV. Each rectangle consists of clustered CSPs based on their properties similarity. Every Rectangle is named in the 2D space as R1, R2, and R3 etc. These are shown in R Tree structured representation in Figure 4.
In this phase, the R Tree is merged with Merkle tree and a hybrid tree is constructed which could provide efficient indexing for fast retrieval of data using R Tree and secure hashed values to provide

integrity through Merkle Tree. This is called MR^{cloud} Tree. The structure of the MR^{cloud} Tree could be seen in Figure 5.

- Every leaf node consists of a CSPs Structure L which holds the <ID of CSP, CSP Property values, H(Properties)>
- Example: For representing CSP11
- L = <CSP11, (10, 1), H (10, 1)>
- Intermediate node I stores < Key_i, H(All its descendent hashed values), $Pointer_i$ >
- Example: For representing R4 I = < (30-50, 2-4), H (H (30, 2) || H (50, 4)), Pointer >
- The Root node holds the Hash of all its descendants and Pointer. It is indicated by P

In this way, the MR^{cloud} Tree should be constructed and signed by the Authority. The Authority has the right to access the database of the CSB. It checks the properties of every CSP stated in MR^{cloud} Tree with the properties signed by the CSP in the Database of CSB by simply decrypting the CSPs profile using its Public key of the CSP. If any difference identified in this, then it will send back the result to the CSB for modification.

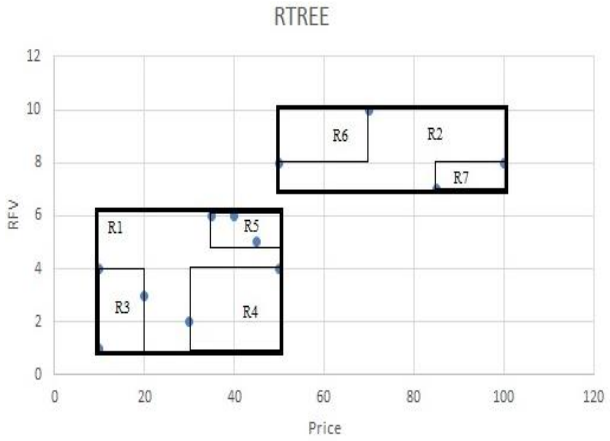


Fig. 3: Example for R tree with MBR.

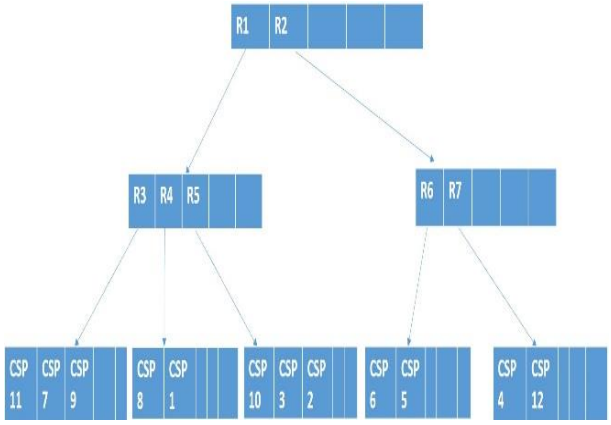


Fig. 4: R Tree of the Sample Data.

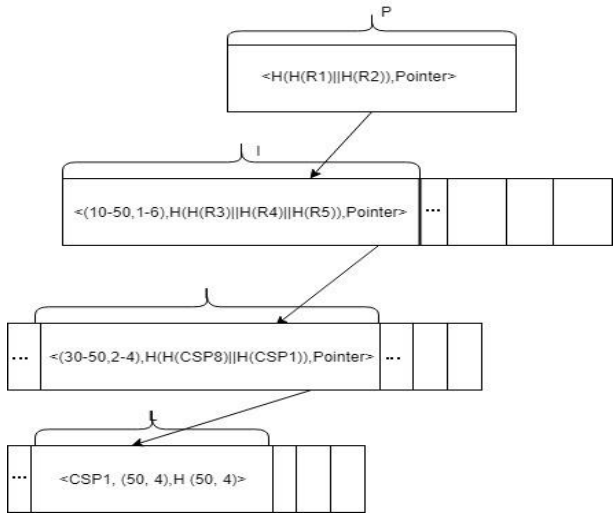


Fig. 5: Sample MR Cloud Tree.

Authority considers the first time as a minor error but if it get repeated frequently then Authority has the right to cancel the registration of CSB and informs all its related CSPs and CSRs regarding the cancellation of license for the bribery CSB. After verifying all the properties of CSPs stated in MR^{cloud} Tree then Authority signs on the Root of the tree with the private key of Authority.

In the general architecture of the Broker, it is noted that after completion of the service transaction the CSR has to give feedback based on the services provided by the CSP. The CSB collects the feedback and calculates Reputation Factor Value (RFV). Here there is a chance of modifying the feedback value of CSP, by CSB to increase or decrease the RFV of particular CSP. To avoid such changes the CSR should submit the feedback by encrypting it using the Blind key. CSB receives the feedback value of a CSP but it can't see the actual value. Simply CSB has to sign on it and send it back to CSR. Then CSR will sign and sends the signed feedback value along with the Blind key to see the value. Using this Blind key the CSB decrypts and appends that value to the profile of CSP and updates the MR^{cloud} Tree. These signed feedback values are also verified by Authority at the time of CSB verification before signing on the root of MR^{cloud} Tree. It sends back that signed MR^{cloud} Tree to the CSB for continuing business and a copy of it is maintained for further reference.

5.2. Query process

CSB collects the request from CSR and runs the Service selection algorithm as a query processing on the MR^{cloud} Tree. It has three steps.

Query pre-processing: CSB examines the query elevated by CSR. In this process, it checks whether query consists of a range of values for all the properties in the database. If any property value is found missing then min and max values of that property are to be considered as a query range for that property.

Selection: All rectangles that intersect the Query region must be retrieved and examined by using a simple recursive procedure that starts at the root node following the pointer to traverse the tree. When a Query Rectangle intersects with a node then that node is processed by retrieving all the rectangles stored in it. If the node is an internal node then the subtrees corresponding to the retrieved rectangles will be searched recursively. Otherwise, the node is a leaf node and the retrieved rectangles are reported.

Single Provider Selection: The stated algorithm reports set of Providers who could satisfy the Query. Among these, single CSP would be selected by considering its RFV value. If the provider is found to have high RFV satisfying all other requirements specified in Query then it would be elected and recommended to the CSR.

For example in the selected data, the range query is to extract the Cloud Service provider who could provide a service in the Price range of 30 to 45 and RFV in the range of 1-4. In general, these ranges make the boundaries to the Query Rectangle which is

shown in Figure 6 with letter Q. It extracts the CSPs in R4. The list of CSPs in R4 is CSP8 and CSP1. After retrieving this set of CSPs select the one which is more overlapping with the Query. The resultant Cloud Service provider is CSP8. In this sample data, two properties for each CSP are considered. But the database consists of a number of properties. In that case all the properties except RFV are to be considered. RFV is generated by CSB based on the feedback given previously. So it is the property which is used to select the one among all similar properties of CSP which is mapped to query.

After the selection of the CSP, CSB has to generate a Proof message. To generate a proof message for Merkle R ^{cloud} Tree, boundaries are to be given to the extracted result. For each and every entry in I (Intermediate Node) must be included in Proof Message. Along with it, Broker appends the parent node and hash values of non-parent nodes. This process would be applied to all the antecedent nodes of I up to the root node. As per the above example, the Proof Message consists of Hash (CSP1), Hash (R4), Hash (R3), Hash (R5) and the Hash (R2) which is the Root node and signed by the Authority. The CSB returns the recommended CSPID, Properties of that CSP, and Proof Message of Merkle R ^{cloud} Tree to the CSR.

5.3. Decision verification

This could be done by CSR itself or by Authority. Some of the CSRs may not be interested to verify the correctness of the CSB on their own to save their business time. In that case, it sends the request to the Authority by providing details like CSB ID, suggested CSP ID, given Query, properties of that CSP and Proof Message.

The four cases would be verified in this phase as follows:

Case1: Authenticity of the CSPs Profile

It could be verified by Authority or CSR by simply inserting Hash (CSP Profile) in the Proof Message and calculating hashes along the way up to the root. Hash (root) is compared to the Proof Message root. If both are the same, it shows the authenticity of the CSP, otherwise, it is considered as a deceit.

Case2: Satisfiability of recommended CSP

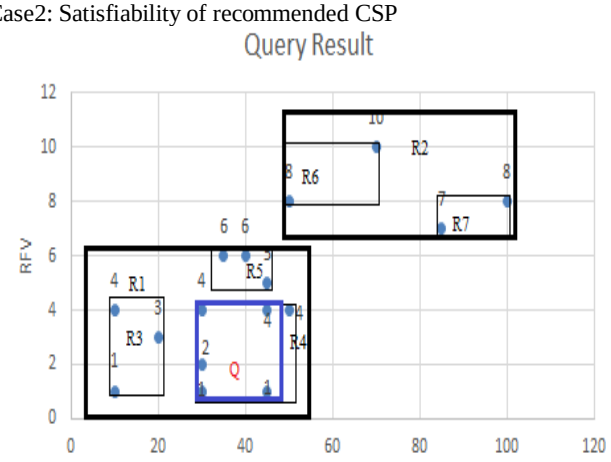


Fig. 6: Range Query on R Tree.

Table 2: Top Providers in the Market

S.No	Provider Name	S.No	Provider Name
1	Amazon EC2	7	OpSource
2	BitRefinery	8	Rackspace
3	GoDaddy	9	ReliaCloud
4	GoGrid	10	Softlayer
5	Hosting.com	11	Terremark*
6	NephoScale		

The satisfiability would be checked by the CSR itself, whether the recommended CSP Properties satisfies the requirements of CSR specified in the Query.

Case3: Integrity of CSP

The integrity of CSP would be verified by the Authority. If any change in the value of RFV then the hash of root will not match with the root node of Proof Message.

Case4: Completeness of the query

This would be verified by Authority only. The Authority verifies the completeness by considering the given boundaries in the Proof Message. If the CSB cheats by leaving some of the qualifying CSPs in the leaf node then it generates the wrong Hash value in the parent node. As per the stated example, CSP1 properties are verified. Hash (CSP1) concatenated with Hash (CSP8) to generate Hash (R4) in the parent node. If this Hash value is not matched with the Hash value given in Proof Message at that corresponding entry indicates the misbehavior of CSB.

6. Performance analysis

6.1. Experimental setup

In this section, the performance of our BDVS scheme with the state-of-the-art MB^{cloud} Tree in all three phases is compared. This approach was implemented by using R Studio. SHA256 was used for Hashing and all the signatures were done by using the RSA Algorithm. All the experiments were conducted on Windows 8 Operating System with 1.7GHZ Intel Corei5 and with 8GB RAM.

6.2. Data preparation

To construct the Dataset, the properties of top 11 Cloud Service Providers in the Marketplace listed in Table2 and Table3 was collected. Only eleven properties which are specific to the task were considered. Random 5 queries on each of the CSP, collected recommendation values, and calculated RFV were applied. Altogether the number of properties considered was 12. These twelve properties were consisting of different types of values. All these properties were categorized and normalized. Pricing Scheme value Pay as you go is replaced by 0 and Monthly by 1. Price property value was normalized into 1 to 100 ranges. Datacenters property value was normalized into 1 to 10 ranges. Certifications property value Yes was replaced by 1 and No by 0. Scale up property value Yes was replaced by 1 and No by 0. Scale-out and Free Tier properties were also considered in the same way. Support property value was categorized into Poor, Average and Extensive. These were numbered as 0, 0.5, and 1. Monitoring and APIs were also categorized in the same way. Oss property values were normalized in the range of 1 to 10. Then a random generator was applied to generate a subset of possible combinations of the property values for new CSPs and the outliers were removed. Only 5000 of them was used as a CSP Dataset.

Table 3: List of Properties

S.No	Property Name	S.No	Property Name
1	Pricing Scheme	9	Monitoring
2	Price	10	APIs
3	SLA	11	Free Tier
4	Datacenters	12	Oss
5	Certifications	13	Instance Types
6	Scale Up	14	Data Transfer out (/GB)
7	Scale-Out	15	Data Transfer in (/GB)
8	Support		

Table 4: List of R Libraries

S.No	Library	S.No	Library
1	Tictac	9	Devtools
2	sodium	10	Rcpp
3	PKI	11	Rtree
4	Readxl	12	data.tree
5	RMySQL	13	ggplot2
6	Digest	14	Plotrix
7	Dplyr	15	Btree
8	rbenchmark	16	SP

6.3. Performance evaluation

The sample dataset size of 500, 1000, 1500, 2000, 2500, 3000, 3500, 4000, 4500, and 5000 was considered for performance evaluation. The list of R libraries used for this proposed work is presented in Table 4. Time taken for the construction of MR^{cloud} Tree is quite less than the time taken for the construction of MB^{cloud} Tree which is shown in Figure 7. Even though the difference is quite less the MR^{cloud} Tree handles Multidimensional data whereas MB^{cloud} Tree handles single dimensional data. Time taken for processing the query on MR^{cloud} Tree is less than the time taken for processing the query on MB^{cloud} Tree which is shown in Figure 8. This is because, in MB^{cloud} Tree, every query was normalized into a single dimension. Time taken for verification on MR^{cloud} Tree was quite less than the time taken for verification on MB^{cloud} Tree which is shown in Figure 9. The difference was very less but MR^{cloud} Tree worked accurately by considering all the dimensions. After comparing the performance of MR^{cloud} Tree with the state-of-the-art, it provided good results.

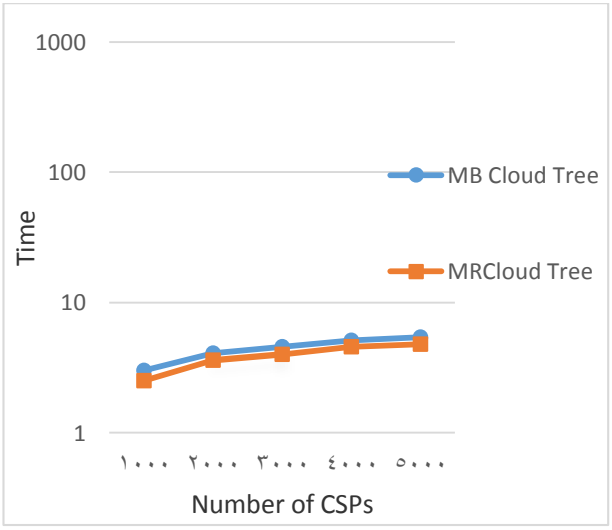


Fig. 7: Structure A Database.

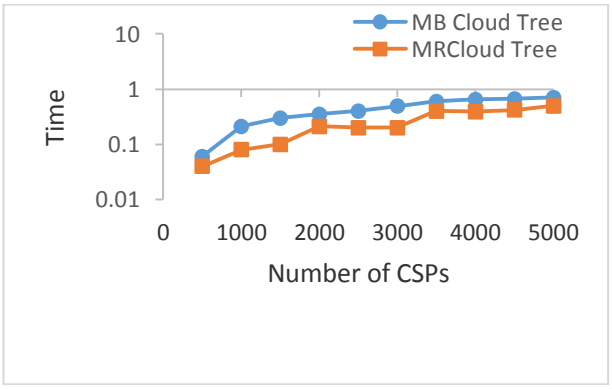


Fig. 8: Query Process.

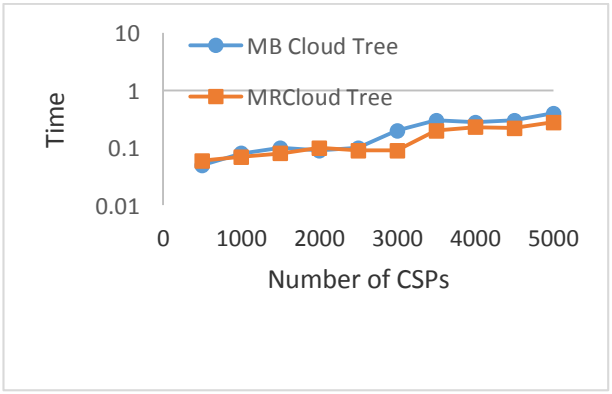


Fig. 9: Decision Verification.

7. Conclusion

In this paper, a Broker Decision Verification System (BDVS) scheme is proposed which provides a facility to the Cloud Service Requester to check the correctness of the Cloud Service Broker. The proposed technique MR^{cloud} Tree works on multidimensional data. This technique is compared to the MB^{cloud} Tree to show the performance of the proposed technique in terms of time in milliseconds. Through this methodology, the Cloud Service Requester could identify the misbehavior of Cloud Service Broker if any. In this work, Broker suggests a single Cloud Service Provider to the Requester which would be done by the RFV property of CSP. In future, this work can be extended by considering Merkle Patricia Tree which would be advanced to the Merkle tree.

References

[1] Hogan M, Liu F, Sokol A, and Tong J: Nist cloud computing standards roadmap-version 1.0. Tech.rep.2011. http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909024 NIST Special Publication 500–291.

[2] Press Releases. (n.d.). Retrieved from https://www.xo.com/about-xo/news-events/press-releases/xo-study-multicloud-strategy.

[3] Cloud Services Brokerage Market by Service Type (Catalog Management, Workload Management, and Operations Management), Platform (Internal Brokerage, External Brokerage), Deployment Model, Organization Size, Vertical, and Region - Global Forecast to 2023. (n.d.). Retrieved from https://www.marketsandmarkets.com/Market-Reports/cloud-brokerage-market 771.html?gclid=Cj0KCQjwnNvaBRCmARIsAOfZq-W1tyNUSz9ohkH4jvTgB5VMaYU5Q_88_ff-BNRk4DAmRVRi_EUd0aAmFNEALw_wcB.

[4] Li, J., Squicciarini, A. C., Lin, D., Sundareswaran, S., & Jia, C. (2017). MMBcloudTree: Authenticated Index for Verifiable Cloud Service Selection. IEEE Transactions on Dependable and Secure Computing, 14(2), 185-198. https://doi.org/10.1109/TDSC.2015.2445752.

[5] Sianipar, J. H., & Meinel, C. (2015). A verification mechanism for cloud brokerage system. 2015 Second International Conference on Computer Science, Computer Engineering, and Social Media (CSCESM). doi:10.1109/cscesm.2015.733188.

[6] Sundareswaran, S., Squicciarini, A., & Lin, D. (2012). A Brokerage-Based Approach for Cloud Service Selection.2012 IEEE Fifth International Conference on Cloud Computing. https://doi.org/10.1109/CLOUD.2012.119.

[7] Koneru, A., & Latha, M. S. (2016). Three-tier architecture: To select CSP through Cloud Service Broker in the multicloud environment. 2016 International Conference on Inventive Computation Technologies (ICICT). https://doi.org/10.1109/INVENTIVE.2016.7823236.

[8] Koneru, A., & M, S. (2017). CLOUD SERVICE BROKER: SELECTION OF PROVIDERS USING DTRFV EVALUATION. Journal of Theoretical and Applied Information Technology, 95(15), 3551-3559. Retrieved from http://www.jatit.org/volumes/Vol95No15/14Vol95No15.pdf.

[9] Huang, K., Zhang, X., Sun, L., & Wang, X. (2017). DLSBD-MHT: Dual-level source-based deduplication with Merkle-Hash-Tree for big data.2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA) https://doi.org/10.1109/ICBDA.2017.8078840.

[10] Hadjieleftheriou, M., Manolopoulos, Y., Theodoridis, Y., & Tsotras, V. J. (2017). R-Trees: A Dynamic Index Structure for Spatial Searching. Encyclopedia of GIS, 1805-1817. https://doi.org/10.1007/978-3-319-17885-1_1151.