CHAPTER 5

INCLUSION OF MAPREDUCE PROGRAMMING MODEL IN CLOUD BROKER FRAMEWORK TO EVALUATE THE TRUST LEVEL OF CLOUD SERVICES

5.1 INTRODUCTION

Nowadays, several cloud service providers have emerged for offering different kinds of services to the cloud users. In such circumstances, trust level of service provider is also emerged as an important issue. A trust (Qu & Buyya 2014) is positive credentials of a service with respect to the QoS factors such as availability, reliability, scalability, privacy, portability, and security. All these factors are most desirable ones and considered while availing the services from the cloud environment. The trust of the services has been either directly proposed by the service provider in terms of service level agreements or indirectly by the feedback responses observed from the cloud user.

The SLA is a formal document about the lifetime of the services, various qualities of service factors, responsibilities of the involved parties. Through this document, the ambiguity of the cloud users’ expectation becomes resolved and the fortification of the service access is achieved. In feedback-based system, the feedbacks given by the respective cloud users are recorded with the appropriate symbols for the finding of its ratings. Many service providers’ experimentation about their trust level is done through the developed web portal. In another scenario, there are some individuals or
groups dubbed themselves as an ‘opinion leader’, a third-party trust evaluators (Chiregi & Navimipour 2016) for deciding the service trust by using ‘troll commands’. By this way, a new cloud user can be influenced to go with a specific cloud service provider. The various types of generally adopted trust level identification techniques are shown in Figure 5.1.

Figure 5.1 Various trust evaluation concepts

Though many researchers attained the trust levels by considering the service provider’s SLAs, user’s feedback, and third parties (broker) opinion, it becomes mandatory to justify the results in terms of its observation.

In SLA based trust evaluation, the service provider is the sole owner and responsible to clearly state the functionalities about the proposed
service. In a cloud environment, the readability, accessibility, and understanding of the SLA is not a simple process. Even though SLAs are adapted to a certain common standard, the most functionality is the documentation about the service and its accessing principle. Through the SLAs, the cloud user can exhibit the service details and focus the agreement towards the business process. Therefore, the trust details are explored as a part of this document.

The opinion from the third parties such as broker (termed as opinion leaders) is also playing a major role in cloud computing by posting their polls about the providers and their services. The comments about the services and its utilization level by various cloud users are considered while calculating the trust values. Though the opinions of different cloud users are considered, it is impossible to match them with all sorts of QoS factors. Some users may be happy with availability and disappointed with security, and others may be vice-versa. Finally, the opinion leaders could not conclude with the partial comments/trolls by the user.

In a feedback based trust evaluation system, all the research works are attempting to record the cloud user’s feedback with the ratings such as good, average, and poor. There were no solutions for the massive number of feedbacks and its processing strategy.

5.2 TOWARDS A BIG DATA FRAMEWORK IN CLOUD SERVICE TRUST EVALUATION

The existing database processing and data mining systems are limited to process certain range of data volume. In a typical cloud computing scenario, the usages of above such normal tools are not sufficient to provide expected outcome in data analysis. With the rapid increase of cloud user’s feedback, it is appropriate to process the huge volume of feedback with the
aid of big data framework. Thus, the notion of big data analytics based on MapReduce framework has been proposed for handling huge volume of user feedbacks about the availed cloud services. The programing paradigm of MapReduce influences the big data platform for effective handling and processing of large volume of structured and semi-structured data.

As a functional programming model, MapReduce (Dean & Ghemawat 2008) process semi structured and unstructured data through Map and Reduce functions. The Map function takes a set of input data and converts it into the form of key-value pairs. And the key-value pairs are sorted for grouping and reduced further to output as the key-value pairs. MapReduce processes the huge volume of data stored in the distributed file system, namely HDFS.

Hadoop has a distributed file system and data processing engine that is designed to handle extremely high volume of data in any structure. It abstracts and facilitates the storage and processing of large volume of structured and unstructured data using simple programming model such as MapReduce Programming. HDFS has a master/slave architecture contains Name node and Data node. The namenode (master) manages the file system namespace and determines the mapping of data chunks (blocks) into data node. A typical data node (Slave) performs read and write requests initiated by the client’s request.

The MapReduce engine is designed to run on top of HDFS as a computational model with the aid of two daemons namely ‘Job tracker’ and ‘Task tracker’. The job tracker receives the job submitted by the client application and instructs the task tracker to run and monitor tasks.
5.3 CLOUD SERVICE TRUST EVALUATION FRAMEWORK

The proposed big data analytics based cloud service trust evaluation framework consists of three important components namely Customer feedback input portal, inclusion of MapReduce programming for processing the QoS parameters, Intelligent Cloud Broker (ICB) as shown in Figure 5.2.

![Diagram of Cloud Service Trust Evaluation Framework](image)

**Figure 5.2 An intelligent cloud broker (ICB) with MapReduce framework**

In this work, an intelligence aspect has been instituted in the proposed broker by incorporating the following features; i) Integration the Hadoop’s MapReduce, (ii) Design of Fuzzy Inference System, (iii) Decision Making System. With the inclusion of above features, the intelligent cloud broker enriches the functionalities such as effective pre-processing of cloud...
user’s feedbacks in terms of QoS parameters and estimating the trust level of services.

The process flow diagram of the proposed architecture is shown in Figure 5.3. At first, the broker allows the authenticated users to post their feedbacks about the services. Further, it initiates the MapReduce framework to pre-process the feedback. Finally, it constructs a Fuzzy Inference System (FIS) and Decision-Making process to evaluate the trust level.

**Figure 5.3 Process flow of the proposed architecture**

The following subsection elaborates about FIS and Decision Making system on the basis of underlying algorithms.
5.3.1 **Pre-processing of User Feedback by MapReduce**

The cloud user’s feedbacks about the availed services with respect to the predefined QoS factors (Availability, Reliability, Portability, Scalability, and Privacy) are processed by the Hadoop’s MapReduce programming model. This process consolidates the feedbacks in terms of *key-value* pairs, which could be easily manipulated for the next phase of the proposed system. The process of MapReduce is shown in Figure 5.4.

![MapReduce workflow](image)

**Figure 5.4 MapReduce workflow**

5.3.2 **Map and Reduce System**

The proposed system has devised the ‘Mappers’ and ‘Reducers’ for the pre-defined QoS features. Consider a service ‘$S_i$’, the cloud users’ feedback with respect to the predefined QoS factors are represented in Table 5.1.
Table 5.1 Cloud user feedback for the service ‘S1’

<table>
<thead>
<tr>
<th>QoS Factors/ Cloud User</th>
<th>Availability</th>
<th>Reliability</th>
<th>Portability</th>
<th>Scalability</th>
<th>Privacy</th>
</tr>
</thead>
<tbody>
<tr>
<td>CU1</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
</tr>
<tr>
<td>CU2</td>
<td>VeryGood</td>
<td>Good</td>
<td>Average</td>
<td>Poor</td>
<td>Good</td>
</tr>
<tr>
<td>CU3</td>
<td>Excellent</td>
<td>VeryGood</td>
<td>Good</td>
<td>Average</td>
<td>Average</td>
</tr>
<tr>
<td>CU4</td>
<td>Good</td>
<td>Poor</td>
<td>Excellent</td>
<td>Good</td>
<td>Good</td>
</tr>
<tr>
<td>CU5</td>
<td>VeryGood</td>
<td>Excellent</td>
<td>VeryGood</td>
<td>VeryGood</td>
<td>Good</td>
</tr>
</tbody>
</table>

5.3.2.1 Mapper Phase

As per the given QoS factors, ‘Availability Mapper’, ‘Reliability Mapper’, ‘Portability Mapper’, ‘Scalability Mapper’, and ‘Privacy Mapper’ for deriving the solution about the service are formulated. All the mappers are having the ‘Map’ function to process the recorded feedback and produce the appropriate key-value pairs. From Table 5.1, the feedbacks about the service ‘S1’, the designated mappers have produced the following result.

\[ A_Mapper = \{(\text{Good}, 2), (\text{VeryGood}, 2), (\text{Excellent}, 1)\} \]

\[ R_Mapper = \{(\text{Good}, 2), (\text{VeryGood}, 1), (\text{Poor}, 1), (\text{Excellent}, 1)\} \]

\[ P_Mapper = \{(\text{Good}, 2), (\text{Average}, 1), (\text{Excellent}, 1), (\text{VeryGood}, 1)\} \]

\[ S_Mapper = \{(\text{Good}, 2), (\text{Poor}, 1), (\text{Average}, 1), (\text{VeryGood}, 1)\} \]

\[ P_Mapper = \{(\text{Good}, 4), (\text{Average}, 1)\} \] (5.1)
5.3.2.2 **Sorting and Grouping Phase**

After the mapping process, each mapper has the appropriate key-value pairs. Next, sorting and grouping phase organizes the *key* in an ascending order and put them together in a husk. The QoS factors and their corresponding keys are sorted and grouped for the appropriate reduce phase and are shown below.

\[
\text{Sorting} \_\text{and} \_\text{Grouping (Mappers)= } \\
\{ (\text{Average}, 1,1,1), (\text{Excellent}, 1,1,1), \\
(\text{Good}, 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1), (\text{Poor}, 1,1), \\
(\text{VeryGood}, 1,1,1,1,1) \} \tag{5.2}
\]

5.3.2.3 **Reducer Phase**

As a final step, the output of all reducer is aggregated to prepare a final key-value pairs. These pairs are the final output of the MapReduce. As per Table 5.1, the outcome of the designed MapReduce framework looks like:

\[
\text{MapReduce} \_\text{output}= \{ (\text{Average}, 3), (\text{Excellent}, 3), \\
(\text{Good}, 12), (\text{Poor}, 2), (\text{VeryGood}, 5) \} \tag{5.3}
\]

Through this processed data, the cloud broker can rate the service for the future reference or recommend a service to the new cloud user. The pseudo-code representation for MapReduce task is given in the algorithm.
Algorithm 1: Mapping_and_Reduction

Input: User_Feedbacks $A_v, R_e, P_o, S_c, P_r$; Service_Rating Poor, Average, Good, VeryGood, Excellent

Output: Mapper_Output $O_m$; Reducer_Output $O_r$; MapReduce_Output $O_{final}$

1. Read the input QoS values with its Feedback data
2. Do for all Input_Feedback
3. If $A_v, R_e, P_o, S_c, P_r$ ! = NULL
4. Invoke Map () function
   4.1 For each Feedback_Values CALL Map ()
   4.2 Find out $O_m$
5. Do Sorting_and_Grouping (OUTPUT (Map ()))
6. Perform Reduce operation on all Map function output
   6.1 For each mapper Map () Do all Reduce ()
   6.2 Find out $O_r$
7. Prepare the final MapReduce output $O_{final}$

5.3.2.4 Fuzzy Inference System

The pre-processed feedback values with respect to the predefined QoS factors from the MapReduce are considered as fuzzy variables for the design of Fuzzy Inference System (FIS). Accordingly, the broker defines five membership functions such as Poor, Average, Good, VeryGood, and Excellent. Each membership function are assigned with appropriate fuzzy weights ranging between 0 and 1. The Box-plot analysis is applied on the fuzzy variables (QoS factors) to fix the weights is shown in Table 5.2.
Table 5.2 Important linguistic variables with its assigned weights

<table>
<thead>
<tr>
<th>Membership Function of QoS factor</th>
<th>Weightage</th>
<th>Membership Function of QoS factor</th>
<th>Weightage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poor</td>
<td>0 - 0.2</td>
<td>VeryGood</td>
<td>0.5 - 0.8</td>
</tr>
<tr>
<td>Average</td>
<td>0.1 - 0.4</td>
<td>Excellent</td>
<td>0.7 - 1</td>
</tr>
<tr>
<td>Good</td>
<td>0.3 - 0.6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The membership functions are created only for the customer convenience and converted to its equivalent crisp values (weight) with the Fuzzification process. The uses of fuzzy linguistic variables play a major role in the fuzzification of user’s feedbacks for the development of fuzzy inference system. Accordingly, defuzzification is applied to transform the aggregated fuzzy quantity to a precise quantity for the user understanding and the same is used to make decision about the trust value of a service. MATLAB provides the facility to create the membership functions with appropriate weights. Using these functions, a system can be developed to deal with the input values against the output. In addition, fuzzy rule-editor simplifies the decision making process by constructing the inference rules as per the input details. As a result, the decision about the service trust can be easily identified in terms of membership degree (Figure 5.5).

**Figure 5.5 Membership function from MATLAB**
5.3.3 Decision-Making System

Even though the developed Fuzzy Inference System (FIS) generates the output with respect to the submitted input QoS factors, a decision about the service-trust needs to be finalized with a well-developed method. With such a technique, one should consider the actual input given by the user, the pre-processed values of the MapReduce system, and the final value of the FIS in terms of linguistic has been taken into the consideration for making decision about the user’s feedback. The following algorithm depicts the decision-making process for the evaluation of service trust.

Algorithm 2: ICB-Decision-Making

Input: Cloud_user $C_u$; Mapper_Output $O_m$; Reducer_Output $O_r$; MapReduce_Output $O_{final}$

Output: Trust_of_Service $ToS$; Recommended_Status $R_s$

1. Check the credibility of the $C_u$ before their feedback analysis
2. Do verification with cloud_user’s log of ICB
   2.1 If $C_u$ is honest then record the feedback
   2.2 Else discard their response
3. Call Mapping_and_Reduction() to preprocess the customer feedback
4. Invoke FuzzyInferenceSystem to generate fuzzy output $Fis_o$
5. Perform FuzzyDecisionMaking with the generated $Fis_o$
6. Publish the service trust value $ToS$
7. Update the Recommended_Status $R_s$ for new cloud users
5.4 EXPERIMENTAL RESULTS

The proposed trust evaluation model has been experimentally validated and its performance is presented in this section.

5.4.1 Experimental Setup

The proposed ICB consists of four major components for evaluating the service trust. To begin with, a .NET based end-user portal for obtaining user’s feedback about the services has been created. With the aid of Oracle’s VirtualBox 5.1, a single node cluster setup with Apache Hadoop Programming model for the demonstration of MapReduce process is created.

The evaluation is conducted with amazon reviews, which contains 34,686,770 reviews from 6,643,669 users about the 2,441,053 products. Then, MATLAB is used to design a Fuzzy Inference System with the membership functions for the QoS parameters. These membership functions have been explored with the help of Fuzzy Rule Editor for the generation of appropriate rules to predict the trust level. Thus, the intelligent cloud broker plays a vital role in the decision-making process to consider the top most services according to its trust value.

As per the experimental setup, a sample of 10 user’s reviews about a single service from the dataset is considered to test the system. Figure 5.6 shows the result of the MapReduce programming model by consolidating the feedbacks with the admissible rating from 0 to 4 (Poor=0, Average=1, Good=2, VeryGood=3, Excellent=4) with respect to the service QoS metrics.
Figure 5.6 MapReduce output of the service QoS factors

Assume that, if there are ‘n’ cloud users availing the service ‘S₁’ and feedbacks are f₁, f₂, f₃, … fₘ, then the output of MapReduce, MRop is formulated as follows.

\[ \text{MRop} (S_1) = \sum_{i=1}^{n} (f_1, f_2, f_3, \ldots f_m) \]  \hspace{1cm} (5.4)

The above equation in terms split into two phases to perform the map and reduce process which are shown in Equations 5.5 & 5.6 respectively.

\[ \text{MapperResult} = \text{Keyvalue} (\text{Feedback} (\text{Poor, Average, Good, Very Good, Excellent})) \] \hspace{1cm} (5.5)

\[ \text{ReducerResult} = \text{Keyvalue} (\text{MapperResult} (\text{Poor, Average, Good, Very Good, Excellent})) \] \hspace{1cm} (5.6)
After pre-processing the user’s feedback, the broker formulate membership functions with the proposed Fuzzy Inference System for the prediction of service trust as per the Equation (5.7). Here, the numerator shows the confluence of QoS factors of a service and the denominator represents the trusted cloud users who recorded their feedbacks after availing the services.

\[
FIS_{Output} = \frac{\|\bar{A}_{val} \cap \bar{R}_{el} \cap \bar{P}_{ort} \cap \bar{S}_{ca} \cap \bar{P}_r \|}{\|\bar{N}_{of\text{GenuineCloudUser}}\|}
\] (5.7)

5.4.2 Observation Analysis

In the proposed system, two observations are adopted to illustrate the trust evaluation process.

5.4.2.1 Observation I

During the first experiment, executed the trust evaluation of 10 selected services by considering provider’s reputation information (Nagarathna et al. 2012) and fuzzy based user’s feedback (Qu & Buyya 2014) which are shown in Figure 5.7. It shows that the fuzzy based feedback system scores better in terms of trust level when compared to the reputation based trust evaluation method. Then, the proposed trust evaluation method based on ICB is applied to attain the significant improvement in trust level (Figure 5.8).
Figure 5.7 Service trust evaluation from the existing methods

Figure 5.8 Service trust evaluation using ICB
5.4.2.2 Observation II

Though the fuzzy based trust evaluation (Qu & Buyya 2014) method scores better results compared to reputation based method (Nagarathna et al. 2012) and other feedback based methods (Wu et al. 2013; Assemi & Schlagwein 2016), it effectively deals only with the minimal set of user feedbacks. When the user’s feedback got increased, the performance of fuzzy based method significantly suffered. With the inclusion of big data framework, effective pre-processing of feedback has been carried out and performance has been increased. A typical increase in performance for the 5000-user’s feedback by the proposed framework is shown in Figure 5.9 with existing methods (Nagarathna et al. 2012; Qu & Buyya 2014).

![Figure 5.9 Experiments on system efficiency with limited user](image)

By extending the user feedback to 50000, the proposed method provides increased system efficiency compared to the counter parts that are shown in Figure 5.10.
5.5 CONCLUSION

In this work, a novel method for cloud service trust by considering the user’s feedback is proposed. The feedbacks about the service’s QoS factors are recorded for pre-processing purpose. With the inclusion of MapReduce programming model, the user’s feedbacks are simplified. The fuzzy inference system of the proposed broker performs the trust level prediction with respect to the predefined QoS factors such as availability, reliability, portability, scalability, and privacy. Then, the decision-making process of the broker concludes the service selection with the highest trust value. The conducted experimental results clearly show that the proposed trust evaluation framework scores better results and enhances the service selection while processing huge number of feedback about the availed services. The next chapter of the thesis describes about the QoS prediction and recommendation of services through the broker which performs on the basis of service contexts.

Figure 5.10 Experiments on system efficiency with increased level of user