De-centralized Reputation-based Trust Model to Discriminate Between Cloud Providers Capable of Processing Big Data

Hadeel T. El Kassabi[1,2], Mohamed Adel Serhani[2]
[1]Concordia Institute for Information Systems Engineering,
Concordia University, Montreal, QC, Canada
h_elkass@encs.concordia.ca
[2]College of Information Technology, UAE University,
Al Ain, UAE
{htalaat, serhanim}@uaeu.ac.ae

***Abstract*—Trust and reputation systems represent a significant trend in decision support including selection of best match cloud providers to process Big Data. Reputation is often considered as a collective measure of trustworthiness based on the referrals or ratings from members in a community. Reputation systems have been applied in various applications such as online service provision. However, reputation models do not reflect user's quality of service (QoS) preferences and thus they might not be satisfied with the recommendations from others. In this paper, we propose a de-centralized reputation-based trust model that incorporates the user QoS preferences to select the best match Cloud Service Provider to process Big Data. Our trust model relies on three multi-attribute decision-making (MADM) methods including Simple Additive Weighting (SAW), Weighted Product Method (WPM), and Technique for Order Preference by Similarity to Ideal Solution (TOPSIS). We conducted several experiments using simulated cloud environment to validate our trust model and assess the three MADM methods. The results show that the proposed model is pliable to users' requirements and efficiently evaluate trust of cloud providers.**

Keywords- *Big Data, Big Data processing, cloud computing, cloud selection, reputation, trust evaluation, trust model, Multi-criteria decision-making, SAW, WPM, TOPSIS.*

I. INTRODUCTION AND BACKGROUND

Cloud computing has emerged as a promising and powerful paradigm for delivering data-intensive, high performance computation, applications and services over the Internet. It has enabled the implementation and success of Big Data, a relatively recent phenomenon describing the abundant data being generated worldwide from scientific discoveries, sensors, social media and other sources. Diverse and competing cloud computing environments have made it challenging to provide an automatic and straightforward way to select a cloud provider that will support Big Data processing and will guarantee user's QoS preferences. Therefore, it is necessary to recommend a trust model to evaluate the QoS of a cloud provider prior to any selection decision.

Trust by nature has many properties among which are Subjectivity, Dynamicity, and Context Dependency [1]. Trust is subjective because it relies on opinion of personal preference of users. Subjective trust assessment is evaluated in literature using probability set and fuzzy set techniques [2]. Alternatively, using self-evidence to evaluate trust objectively raises some challenging issues, of which incompleteness and uncertainty. Dynamicity is another property of trust that is highly dependent on elapsed time, amount of interactions, and variation of physical resource capabilities over time. Hence, periodic re-evaluation of trust is required. Moreover, trust is context dependent since an entity can be trusted in a domain but not in another. Many trust models addressed this property [2], [3], and [4]. However, not all trust models deal with all trust properties, which make them ineffective.

Recently, the number of cloud service providers has increased exponentially making it difficult for the users to choose among them. Assessing trust of cloud providers is important to discriminate between them, especially when dealing with Big Data. In the literature, several trust models are based on reputation information to measure the trustworthiness of a cloud service provider (CSP). However, existing reputation systems are usually centralized, which exhibit some limitations related to their flexibility, dynamicity, heterogeneity and scalability in measuring trust. Reputation carries also another restriction which is utilizing the judgement of other users with respect to service that does not necessarily represent the user's personal QoS preference.

Measuring QoS attributes is not always an easy practice users can do as they might lack the technical knowledge and might have subjective judgments. Additionally, collecting QoS attributes from user is a not a straightforward task because the QoS attributes should match the user application requirements and allow the input of user preference. Existing commercial and noncommercial service selection systems lack support for users with respect to defining the QoS requirements and lack the automation capability for collecting Big Data and cloud quality. Hence, automating the decision-making process of cloud provider selection with an eye toward Big Data processing requirements and user QoS preferences is highly desirable.

In this paper, we propose a de-centralized reputation-based trust model that incorporates the user QoS preferences in CSP selection decision for Big Data processing. This model allows automatic decision-making for selecting the best cloud service provider for Big Data processing using the other's experience in addition to incorporating the user's point of view. In the proposed reputation-based trust model, we formalize the trust as a multi-attribute decision-making problem. We apply three different scoring methods to evaluate the performance of cloud service providers. These methods are Simple Additive Weighting (SAW) Method, Weighted

978-1-5386-1996-4/17 $31.00 © 2017 IEEE
DOI 10.1109/BigDataCongress.2017.41

Product Method (WPM), and Technique for Order Preference by Similarity to Ideal Solution (TOPSIS). We implemented a cloud simulation environment using Java language to test and validate our reputation-based trust model. The implemented experiments, proved that our trust model captures most of trust properties including flexibility, dynamicity, heterogeneity and scalability. We also compared the three proposed MADM methods against a benchmarking reputation method that does not incorporate user preference. The results show that the proposed model is pliable to users' requirements and preference.

II. RELATED WORK

In this section, we survey existing reputation trust models, their characteristics as well as their classification. We also, review computational techniques used to measure trust score that are considered as a key factor in trust evaluation.

A. Reputation Trust Models

Reputation is connected to the concept of trustworthiness, although it is still different [5]. In this paper, we use the definition of reputation by the Concise Oxford dictionary which is: *"what is generally said or believed about a person's or thing's character or standing."* The quality of service provisioning of a CSP as perceived by the users, determines its reputation. Trust can be achieved by a good reputation. However, because trust by nature is subjective, trust can also be given to a CSP based on good self-experience despite having bad reputation. Nevertheless, when the self-experience is absent, trust is usually better be determined by reputation.

A reputation-based trust model relies on the opinions and experiences of other users towards cloud service providers. Many reputation-based trust models were proposed in the literature. We classify these models into service oriented and resource oriented models according to the type of quality attributes used as basis to evaluate the trust score. The service oriented models assess trust according to the Quality of Service guarantee of the cloud provider. But, the resource oriented models depend on the quality resources provided by the cloud and the availability to calculate trust score.

1) Service Oriented

Several service quality-based reputation trust models were proposed in the literature. Authors in [6] proposed a discovery system with a registry which contains a list of service providers along with feedback information from other service providers and users. The trust score is evaluated based on the standard deviation, which is considered to be inversely proportional to trust. Reputation also was used in [7] to evaluate trust for mobile ad-hoc clouds. They use a set of attributes such as availability, neighbor evaluation, response quality and task completeness to evaluate the quality of cloud nodes. Likewise, trust was evaluated in [8] using QoS attributes such as accountability, skills, service reliability, cost, performance, security, privacy and usability.

Moreover, some initiatives proposed exploiting users' experience with the service rather than their opinion about it [9]. The authors calculated trust scores using adaptive modeling algorithms which are: the rough set and induced ordered weighted averaging (IOWA). The rough set allow using objective assignment of QoS attributes. Also, the IOWA operator adapts to the dynamic nature of the cloud as it uses time series for trust evaluation. Also, authors in [10], use multiple QoS attributes to evaluate trust; yet, they manually and uniformly assign the weights which restrain the user to flexibly encompass his/her preferences.

Other initiatives were proposed in the context of Big Data and cloud computing. In [3], the authors proposed a category-based context-aware and recommendation incentive-based reputation mechanism (CCRM) to enhance veracity and protect data against internal attacks. Another approach was introduced in [11] were a dynamic trust model that considers the user preferences and false ratings in trust evaluation. In [12] a resource broker is used to evaluate trust for grid and cloud resources considering user preference. Still, the authors considered simple factors that did not satisfy the complexity of the trust evaluation [9]. A trust framework named TRUSS for cloud service selection is proposed in [13]. The later uses objective and subjective trust evaluation using QoS monitoring and feedback ratings. Also authors in [14] combine objective and subjective models for trust evaluation.

2) Resource Oriented

Few propositions focused on resource quality for trust evaluation. In [15], an IaaS-based trust model was proposed. The model uses parameters such as the processing capabilities of the virtual machines (VMs), i.e., processing speed, fault rate, bandwidth and price for trust evaluation. Nevertheless, the experiments did not compare results to other trust models. Moreover, authors in [16] proposed a trust model to improve file transfers between nodes of a private cloud. The trust score is calculated based on node storage space, the operating system, network bandwidth and processing capacity.

B. Trust Score Computation methods

Trust score is computed in various ways in reputation systems. eBay's reputation forum uses the difference between the number of positive ratings and the number of negative ratings to get a trust score [17]. It is one of the simplest methods but it may lead to ineffective results. Other commercial websites such as Epinions and Amazon, use the average of all the ratings to calculate the trust score. Alternatively, trust scores were also computed using weighted sum as depicted in [14]. Another flavor of weighted sum trust by using the rater's credibility, age and distance between the new and existing ratings as weights. However, the reputation system trust scores are usually calculated by other community members that might have different user requirements, environment or QoS priorities or preferences.

Authors in [18] reviewed other types of computational reputation models such as Bayesian Systems [19], Regression Analysis [20], Belief Models [21], Fuzzy Models [22], [23], [24] and Flow Models [25]. Nonetheless, some of these methods exhibit high algorithm complexity and require extensive time which makes it unsuitable for cloud service provider trust evaluation. In addition, trust computation methods are also associated with the way the trust score and trust attributes are scaled. Trust score can take different scale forms such as binary, discrete, nominal scale, and continuous values [1].

Existing reputation trust models are mostly centralized, non-dynamic in nature and lack the real-time adaptability [26]. These properties do not fit the dynamicity and heterogeneity of Big Data applications and the cloud environment. Other reputation trust models have based their trust score computation on weights that are not necessarily suitable to the user's choice.

III. De-centralized Reputation-Based Trust Framework

In this section, we first describe a mapping scheme to map some Big Data characteristics to their correlated cloud quality attributes, then we depict our trust model framework for selecting a cloud service provider for Big Data processing over cloud. We use the provider's reputation gathered from other users' historical Quality of Cloud Service (QoCS) records. In addition, we consider the user QoCS preference during trust evaluation in order to have favorable and rational score. Figure 2. describes the components of our proposed framework and section B below details the role of each component and their interactions.

A. Trust Attributes for Quality Evaluation

Figure 1. represents the relationship mapping between Big Data properties and cloud quality metrics. Our model use these quality attributes to evaluate the degree of trustworthiness of the cloud providers.

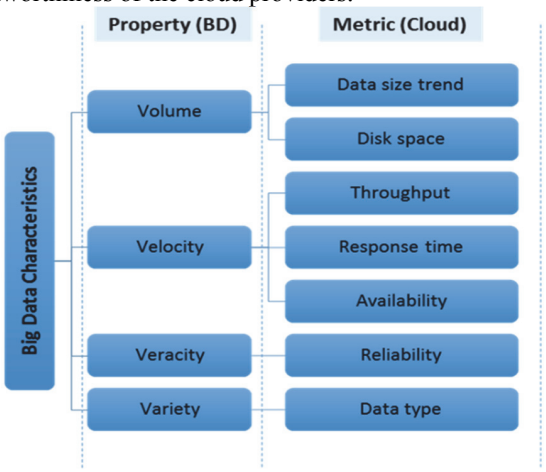

Figure 1. Big Data QoS Attributes

We measure the aptitude of the cloud service provider to process Big Data by considering some key Big Data characteristics including volume, velocity, veracity and variety. Therefore, we select specific Big Data quality attributes which we consider significant factors in cloud provider selection. We propose a mapping scheme for some of these Big Data characteristics to their correlated cloud quality attributes in Figure 1. . As follows:

Volume: We map this characteristic to two main metrics. First, the size of the data to be processed. The class of this attribute (1: low, 2: medium, 3: high, 4: very high). Second, the size of available disk space at the cloud provider (Disk space).

Velocity: is mapped to four metrics; which are resource quality, response time, availability, and throughput. The rest of aforementioned QoCS attributes are measured according to cloud characteristics and behavior observed during interaction communications as follows:

- ***Response time:*** The actual execution time = the time spent between sending a request and receiving the last byte of the response, in milliseconds.
- ***Availability:*** The ratio of the number of received responses to the number of sent requests.
- ***Throughput:*** The number of requests handled per second = (total number of requests / end time − start time) × 1000.

Veracity: is mapped to ***Reliability*** which is the task success ratio measured as the total number of task requests - the number of illegal connections and the number of denial of service incidents /total number of task requests.

Variety: relates to the type of data to be processed. Class 1 refers to structured data, class 2 refers to unstructured data, and class 3 refers to hybrid structured and unstructured data.

B. Reputation-based trust /Community-based trust

In this study, we propose a trust framework which use reputation information within a neighborhood community to evaluate trust of cloud service providers. This framework consists of two main components as shown in Figure 2.

The first component is located at the user side, which is responsible for: 1) collecting preference QoS information from the user, which is performed by ***User QoS Preference*** module, 2) sending reputation request message to neighbors containing personal preference, 3) collecting reputation scores from responding neighbors, which is handled by ***CP Reputation*** Module, and 4) generating an average trust score for each CSP and provides the user with a recommendation of the CSP having the highest score. This activity is the responsibility of the ***Trust*** Module.

The second component is located at each neighbor or community member and is responsible for: 1) monitoring the self-transactions with other CSPs, 2) keeping a history log in a database called ***Local History database***, 2) receiving reputation request messages from other users, 3) generating a trust score for each CSP upon receiving a request message., and 4) sending reply messages containing trust scores to each requesting user.

The trust score for each CSP is generated by applying one of the MADM algorithm proposed in this paper; SAW, WPM and TOPSIS. In this model, the weights, are the QoS preferences which are sent by the requesting user, whereas the attribute values are the CSP performance in each metric which are stored in the history log. This model will be detailed in the next section. Each community member host a ***User Request Handler*** module to process user reputation requests. This module receives request messages from the user containing the preferred QoCS attributes and their weights. When a request message is received, the module generates a trust score for each CSP (*CPscore*) after extracting the information stored in the ***Local History database***. The CPscore is sent in a reply message that to the ***CP Reputation Module*** at the

requesting user. The ***Local History database*** consists of a local history log of communication with CSPs.

Community Management: In our trust model, the trust score evaluation, relies on the CSP's reputation within the community neighborhood. Hence, we should manage trust among community members to establish trust for the reputation information they provide. Community by the definition provided in the Oxford dictionary, is *"the condition of sharing or having certain attitudes and interests in common"*. Hence, community members have mutual interest in providing and obtaining CSP reputation information to and from other community members. Moreover, the community members are motivated to provide reputation information to their neighbors as they can gather neighbors' scores in return and have more comprehensive trust information. Community management has been proposed in [27], [28].

The user will generate messages requesting the CSP scores from the community members based on: 1) user preferred QoCS attributes, and 2) the service history of the community members. To encourage neighbors to provide reputation information, incentives should be provided to them [5], such as receiving CSP reputation scores from the community to help in their own decision-making. The user request message encloses a list of QoCS attributes and their user assigned weights. When receiving a request from a user, the neighboring users perform the following:

1 Calculate the CPScore for each CP using one selected method among the three proposed methods detailed in the next section called ($CPscore_i$) using the weights embedded in the request message.
2 Send a reply message with a list of CP scores to the user who originated the reputation request message.

The local user receiving the reputation reply message parses it to extract the scores for each CSP, as shown in Figure 2. After receiving all the replies, the user calculates the average score of each CSP among the *n* users who replied. Using average in this case is appropriate and proven to be one of the simplest ways to obtain the mean value.

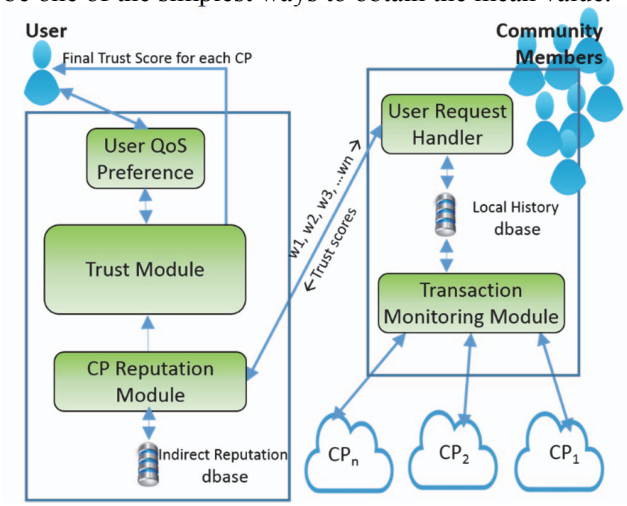

Figure 2. Reputation Trust Model Framework.

IV. Formal Trust Evaluation using Multi Atribute Decision Making Techniques

In this section, we describe our formal trust evaluation model. We formulate our model for trust evaluation as a multi-attribute decision-making model (MADM). We choose three multi-attribute scoring methods for evaluating the trust score for each CSP. These include the Simple Additive Weighting method (SAW), the Weighted Product Method (WPM), and the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS).

These methods follow a similar version of the Simple Multi-Attribute Rating Technique (SMART) described in [29] the follow these steps:

- Step1: Determine the goal which is selecting the most suitable cloud service provider for Big Data processing.
- Step2: Identify the alternatives for evaluation what are the available cloud service providers in the market.
- Step3: Determine the attributes that are used as basis for evaluating the alternatives. For example, throughput, reliability, resource quality, etc.
- Step4: Choose the weights for each attribute. In other words determine the importance of each attribute.
- Step5: Evaluate the score of each alternative using one of the three MADM scoring methods.
- Step6: Analyze the results and reach a decision of the best alternative.

We follow the SMART technique because it is considered the most common method actually used in real, decision-guiding multi-attribute utility measurements [30]. The smart model does not depend on the alternatives and is not affected if more alternatives are added. In the following subsections, we describe the details of each method:

A. Simple Additive Weighting Model

This method is also named as weighted sum model (WSM). It is the simplest and one of the most widely used methods [31]. The rank or trust score is calculated using Simple Additive Weighting (SAW) method with weights assigned for each alternative. The values of the alternatives ranks are used to choose the best alternative [32] [33].

We formulate the score for each cloud provider as a Multi-Attribute Decision Making (MADM) problem [34] wherein the model is expressed as follows:

Step1: Model construction/initialization

$$\mathrm{CP} = \{cp_i | i = 1,2,3, \ldots n\} \quad (1)$$

$$\mathrm{A} = \{a_j | j = 1,2,3, \ldots m\} \quad (2)$$

$$\mathrm{W} = \{w_1, w_2, w_3 \ldots w_m\} \quad (3)$$

$$X = \begin{bmatrix} x_{11} & \cdots & x_{n1} \\ \vdots & \ddots & \vdots \\ x_{1m} & \cdots & x_{nm} \end{bmatrix} \quad (4)$$

where cp_1, cp_2 … cp_n are the possible n alternative cloud service providers available to the user, a_1, a_2,..., a_m represent QoCS attributes (criteria) for example: response time, availability and reliability. w_j is the weight (significance) of the j^{th} attribute and x_{ij} is the performance rating of the i^{th} alternative (cp) with respect to the j^{th} attribute. In this model, a higher score is assigned to the cloud provider with the highest performance rating, which preferably maximizes the j^{th} attribute.

Step 2: Construct the normalized decision matrix. This step is needed to allow comparing attribute values with different scale units. The values here are normalized on a scale from 0 to 1. Some attributes, such as reliability, have preferably high values, whereas others, such as cost, have preferably low values. Thus, to normalize these values easily and fairly, we use Eq. 5 when a high value is preferred (beneficial attribute) and Eq. 6 when a low value is preferred (non-beneficial attribute).

$$r_{ij} = x_{ij} / x_{ij}^{max} \quad \text{(row)} \quad (5)$$

$$\text{Or } r_{ij} = x_{ij}^{min} / x_{ij} \quad \text{(row)} \quad (6)$$

Step 3: Construct the weighted normalized decision matrix. We give a different weight value for each attribute to give different preference of an attributes over other attributes. The user selects the weights based on QoS preferences and the type of Big Data application. We use the following equation to calculate the values of the weighted normalized decision matrix:

$$v_{ij} = w_j * r_{ij}, \text{ s.t.} \sum_{j=1}^{m} w_j = 1 \quad (7)$$

Step 4: Calculate the score of each alternative (cp):

$$score_i = \sum_{j=1}^{m} v_{ij}, \; i = 1,2,3,.., n \quad (8)$$

Step 5: Select the best alternative (CP):

$$CPbestscore = \max_{0 \le i \le n} score_i \quad (9)$$

B. Weighted Product Method (WPM)

Another approach, is to use the weighted product method as a scoring technique. This method was first introduced by Bridgeman [35], it is simple method that is not widely used despite its sound logic [33]. It is introduced as a modification to the SAW method [31]. It does not require normalization because the attributes are multiplied to each other and raised to the weights as an exponent. For beneficial attributes; i.e. the attributes that are better when have higher values; the weight exponent should be positive. Negative weights are given for non-beneficial attributes [33]. The score is then calculated using the following formula [36]:

$$score(cp_j) = \frac{\prod_{i=1}^{m} x_{ij}^{w_i}}{\prod_{i=1}^{m} x_i'^{w_i}} \quad (10)$$

where x_i' value is the highest score of the attribute i among all alternatives (CPs). Using x' allows to limit the resultant score value to be between 0 and 1 instead of using numbers that are greater than 1 because of the exponent property.

C. Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)

The third scoring approach used in this work is the TOPSIS technique proposed by Hwang and Yoon [37]. It is based on the idea of choosing the alternative with the shortest distance from the positive–ideal solution and the longest distance from the negative-ideal solution. Here the ideal solution is the assembled ideal scores of all attributes. The following are the steps required for this method:

Step 1 and ***Step2*** are the same as in the SAW model.

Step3: Identify Positive-Ideal and Negative-Ideal Solutions. The Positive-Ideal solution is the highest performance value for attribute i among all alternatives and is represented as follows:

$$X^* = \{x_j^* | j = 1,2,3, \dots m\} \quad (11)$$

The Negative-Ideal solution is the lowest performance values for attribute i among all alternatives and is denoted as follows:

$$X^- = \{x_j^- | j = 1,2,3, \dots m\} \quad (12)$$

Step 4: Calculate Separation Measures. This is done by calculating the distance of each alternative cp_i from the positive-ideal solution X^* using the n-dimensional Euclidean distance:

$$D_i^* = \sqrt{\sum_{j=1}^{m} (x_{ij} - x_j^*)^2} \quad (13)$$

Where i is the alternative index, and j is the attributes index. Also the separation from the negative-ideal solution X^-, is given by:

$$D_i^- = \sqrt{\sum_{j=1}^{m} (x_{ij} - x_j^-)^2} \quad (14)$$

Step 5: Calculate Similarity Indexes. We calculate the similarity index for alternative cp_i using the following:

$$score_i = \frac{D_i^-}{D_i^* - D_i^-} \quad (15)$$

Where the $0 <= score_i <= 1$. The cp_i with the highest $score$ is selected to process Big Data.

V. EVALUATION

In this section, we describe the communication overhead evaluation and the experiments we have conducted to compare the QoS attributes performance against the three proposed approaches for reputation-based trust model. The purpose of our evaluation is to verify that our proposed trust model performs appropriately CSP selection to process Big Data while capturing key properties of Big Data and incorporating user's QoS preferences.

A. Environment Setup

For our simulation we used the following parameters and machine configuration:

- A computer desktop with Intel CoreTM i7-3770K CPU @ 3.40GHz with Turbo Boast, 32GB of DDR3 RAM, 1TB hard drive, and 64-bit operating system
- Number of clouds: 1 to 50 clouds
- Number of nodes within each cloud: 1 to 100 nodes.
- QoCS attributes: data size, distance, cost, response time, availability and confidence.
- Number of community members: 3 to 100 neighbors.
- Reputation database: 20 interaction log records for each CSP local to each community member.

B. *Simulation Framework*

To test our proposed model, we developed a dedicated simulator using Java. We implemented the modules previously described in Figure 2. mainly, the user modules and the neighbor modules. The modules residing on the user side are described as follows:

User QoS Preference: implemented as Web application that collects the QoS information from the user and generates appropriate a list of QoS attributes and a weight value for each. *Trust Module:* is the main module which retrieves the required QoS information and uses this to evaluate a trust score for each CP and recommends the one with the highest score to the user. It implements all three algorithms explained previously in section 4. C*P Reputation Module:* implemented to simulate the clouds reputation information collection from the neighboring users. *Indirect Reputation database:* is the database that contains the average trust scores collected from all neighbors.

The component which simulates the community member encompasses the following components:
User Request Handler: This module uses the log generated by communicating with simulated cloud objects to populate the *Local History* database then it responds to reputation requests from other users with trust scores for all CPs by analyzing its own *Local History* database and the weights for each QoS attributes extracted from the request messages. *Local History* database contains the quality information of the simulated transactions with other CPs.

C. *Experimental Scenarios*

In this section, we describe the main scenarios we have developed to evaluate our reputation-based trust model.
Scenario 1: we evaluated the proposed MADM algorithms by comparing the different generated ranking results. Figure 3. shows that closer ranking results are produced by SAW and TOPSIS algorithms. However, the WPM algorithm shows a different ranking trend.
Scenario 2: we evaluated the scalability of our trust model where we increase the number of CSPs and measure the QoS of the selected CSP by each of the three MADM algorithms. We measure response time and cost quality attributes for this experiment. Figure 4. shows that response time and cost of the chosen CSP decrease as the number of CSPs increases for all three algorithms. This is because the more CSPs we have the more options we have and eventually more chance to get CSP offering better performance.

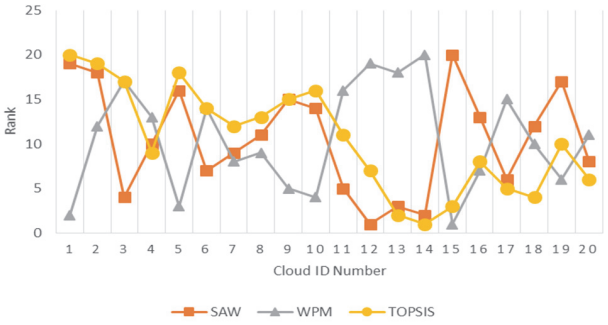

Figure 3. CSP Rank

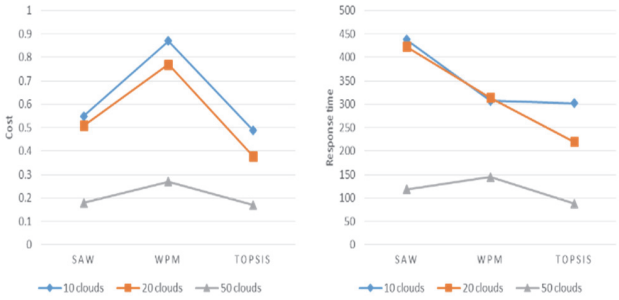

Figure 4. Scalability test of Reputation Trust Model.

Scenario 3: we evaluated the different weights assigned to different QoS attributes and we analyzed the effect of changing the attribute weight on trust scores generated by each of the three MADM algorithms. We chose response time, availability, and cost quality attributes for this experiment. Figure 5. shows that response time of the chosen CSP decreased as its weight value increased for all three algorithms. As the response time was made relatively more important, i.e. was favored over the other attributes for cloud selection, the selected cloud accordingly shows better response time. We can also observe that the selection using SAW algorithm results in a better response time followed by the TOPSIS, then the WPM.

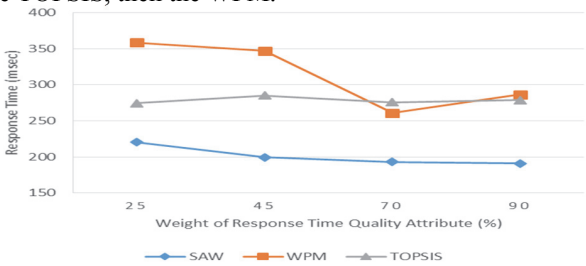

Figure 5. Response Time - Reputation Trust Model.

Figure 6. shows that the availability of selected CSP increases as its weight value increases for all three algorithms. This is because availability was also favored over the other attributes for cloud selection, the selected cloud accordingly shows higher availability. Moreover, we can conclude that the selection based on TOPSIS algorithm results in a better availability followed by WPM and SAW algorithms.

In addition to response time and availability, we also tested the cost quality attribute. As shown in Figure 7. , when the cost was favored over the rest of quality attributes, the selected CSP has a low cost for all three algorithms. The TOPSIS again, gives a better selection results as it shows lower cost values than the other two algorithms.

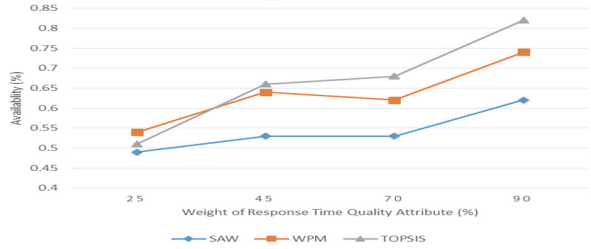

Figure 6. Availability - Reputation Trust Model.

Figure 7. Cost - Reputation Trust Model.

Scenario 4: we benchmark the three scoring algorithms with reputation-based trust score calculated without user preferences. We compare our proposed user preference–based reputation model to a reputation model that doesn't involve the user's preference. The benchmark reputation model (BMRM) hence, doesn't use weights for the attributes, but just provides a local trust score to the user. We chose response time and cost quality attributes for this scenario. First, we give higher weight in favor to cost quality attribute. The test shows that BMRM gives CSP selection with higher cost than the CSPs selected using our proposed model with the TOPSIS giving the best performance as in Figure 8. . Second, we performed the test giving higher weight for the response time. Again, the results show that our proposed model using SAW, WPM, and TOPSIS algorithms perform better than the BMRM as shown in Figure 8. .

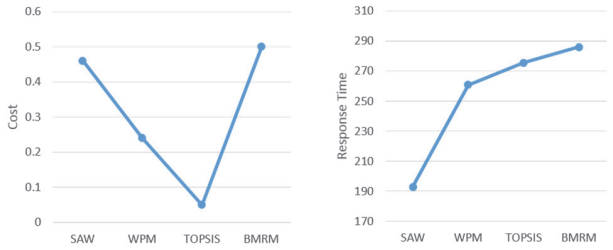

Figure 8. Cost and Response Time Using Different Algorithms.

D. Communication Overhead

In this section, we evaluate the communication overhead for our reputation model by measuring the number and the size of messages exchanged for the purpose of trust evaluation. Our proposed trust model requires messages to be exchanged between the user and the neighboring community members to collect the trust scores. A request message is sent to the neighboring users, and a response message is sent back containing the trust score values for each CSP. The request message contains the QoS attributes names and weights. Thus, we calculate the request overhead using the following equation:

$$reqOverhead = n * [nQoS * (qnSize + qwSize)] \quad (16)$$

where n is the number of neighbors, nQoS is the number of QoS attributes used for trust score evaluation, qnSize is the QoS name size which is about two bytes, qwSize is the weight value of the QoS attribute which has the size of a number i.e. about four bytes.

For the response message, we evaluate the overhead as the number of responses received from neighbors. The size of each message would be measured as follows:

$$respOverhead = n * [nCSP * scoreSize] \quad (17)$$

where *n* is the number of neighbors (the maximum number of responses), *nCSPs* is the number of available CSPs, *scoreSize* is the trust score that has the size of a number which is about four bytes in size.

Figure 9. shows that communication overhead is proportional to the number of selected QoCS attributes and also the number of the community members. For example, the calculated overall communication overhead for 20 cloud providers, 100 active community members and 50 selected QoCS attributes, was less than 40 Kbytes, which is almost negligible.

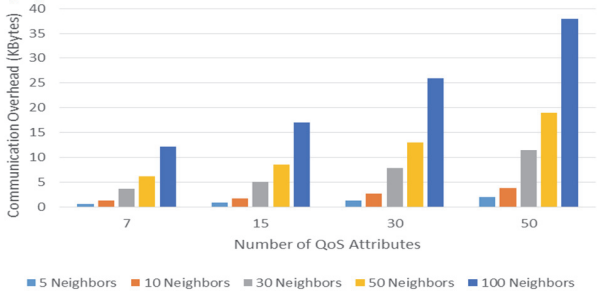

Figure 9. Reputation Trust Model Communicatin Overhead.

VI. Conclusion

In this paper, we proposed a de-centralized reputation-based trust model to support Big Data processing over various cloud providers offering similar services. The model took into consideration the user QoS preferences in calculating trust scores. We also model the cloud selection problem as multi-attribute decision-making (MADM) relying on three trust scoring schemes; SAW, WPM and TOPSIS. In addition, the model captured Big data key characteristics and coped with some key features including flexibility, heterogeneity, and scalability of the studied environment. We conducted a set of experiments using simulated cloud environment we have developed for the purpose of validating our trust model and assessing the three MADM methods. The results demonstrated that our proposed model capture users' requirements and efficiently evaluate trust of cloud providers. Experiments results also showed that TOPSIS model

generally gives better results with respect to tested quality attributes such as response time and cost. As future work, we are planning to extend the model to cope with malicious reputation information. We are also planning to run more extensive experiments using real-world data which reflects other Big Data features.

ACKNOWLEDGMENT

This work is supported by UAE University UPAR Grant # 31T064.

REFERENCES

[1] J.-H. Cho, K. Chan and S. Adali, "A survey on trust modeling," *ACM Computing Surveys (CSUR) ,* vol. 48, no. 2, p. 28, 2015.

[2] A. Kanwal, R. Masood and M. A. Shibl, "Taxonomy for Trust Models in Cloud Computing," *The Computer Journal ,* 2014.

[3] H. Lin, J. Hu, J. Liu, L. Xu and Y. Wu, "A Context Aware Reputation Mechanism for Enhancing Big Data Veracity in Mobile Cloud Computing," in *IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM)*, 2015.

[4] O. Malacka, J. Samek and F. Zboril, "Event driven multi-context trust model," in *2010 10th International Conference on Intelligent Systems Design and Applications*, 2010.

[5] A. Josang, R. Ismail and C. Boyd, "A Survey of Trust and Reputation Systems for Online Service Provision," *Decision support systems,* vol. 43, no. 2, pp. 618-644, 2007.

[6] M. K. Muchahari and S. K. Sinha, "A new trust management architecture for cloud computing environment," in *IEEE International Symposium on Cloud and Services Computing (ISCOS)*, 2012 .

[7] A. Hammam and S. Senbel, "A trust management system for ad-hoc mobile clouds," in *IEEE 8th International Conference on Computer Engineering & Systems (ICCES)*, 2013.

[8] A. Gholami and M. G. Arani, "A Trust Model Based on Quality of Service in Cloud Computing Environment," *International Journal of Database Theory and Application,* vol. 8, no. 5, pp. 161-170, 2015.

[9] X. Li and J. Du, "Adaptive and attribute-based trust model for service level agreement guarantee in cloud computing," *IET Information Security,* vol. 7, no. 1, pp. 39-50, 2013.

[10] H. Kim, H. Lee, W. Kim and Y. Kim, "A trust evaluation model for QoS guarantee in cloud systems," *International Journal of Grid and Distributed Computing,* vol. 3, no. 1, pp. 1-10, 2010.

[11] B. Li, L. Liao, H. Leung and R. Song, "PHAT: A Preference and Honesty Aware Trust Model for Web Services," *IEEE Transactions on Network and Service Management,* vol. 11, no. 3, pp. 363-375, 2014.

[12] "Trust management system for grid and cloud resources," in *IEEE First International Conference on Advanced Computing*, 2009.

[13] M. Tang, X. Dai, J. Liu and J. Chen, "Towards a trust evaluation middleware for cloud service selection," *Future Generation Computer Systems,* 2016.

[14] M. Nitti, R. Girau and L. Atzori, "Trustworthiness management in the social internet of things," *IEEE Transactions on knowledge and data engineering,* vol. 26, no. 5, pp. 1253-1266, 2014.

[15] M. K. Goyal, A. Aggarwal, P. Gupta and P. Kumar, "QoS based trust management model for Cloud IaaS," in *2nd IEEE International Conference on Parallel Distributed and Grid Computing (PDGC)*, 2012.

[16] E. D. Canedo, R. T. de Sousa, R. de Oliveira and F. L. L. de Mendo, "File Exchange in a Private Cloud supported by a Trust Model," in *IEEE International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, 2012.

[17] P. Resnick and R. Zeckhauser, "Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system," *The Economics of the Internet and E-commerce,* vol. 11, no. 2, pp. 23-25, 2002.

[18] J. Guo and R. Chen, "A Classification of Trust Computation Models for Service-Oriented Internet of Things Systems," in *2015 IEEE International Conference on Services Computing (SCC)*, 2015 .

[19] A. Jsang and R. Ismail, "The beta reputation system," in *Proceedings of the 15th bled electronic commerce conference*, 2002.

[20] Y. Wang, Y.-C. Lu, I.-R. Chen, J.-H. Cho, A. Swami and C.-T. Lu, "LogitTrust: A logit regression-based trust model for mobile ad hoc networks," 2015.

[21] A. Jøsang, "A logic for uncertain probabilities," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,* vol. 9, no. 3, p. 279– 311, 2001.

[22] A. M. Hammadi and O. Hussain, "A framework for SLA assurance in cloud computing," in *IEEE 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, 2012.

[23] D. Chen, G. Chang, D. Sun, J. Li, J. Jia and X. Wang, "TRM-IoT: A trust management model based on fuzzy reputation for internet of things," *Comput. Sci. Inf. Syst.,* vol. 8, no. 4, pp. 1207-1228, 2011.

[24] J. Sabater and C. Sierra, "Reputation and social network analysis in multi-agent systems," in *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, 2002.

[25] L. Page, S. Brin, R. Motwani and T. Winograd, "The PageRank citation ranking: bringing order to the web," Technical report, Stanford Digital Library Technologies, 1999.

[26] S. Nusrat and J. Vassileva, "Recommending services in a trust-based decentralized user modeling system," in *International Conference on User Modeling, Adaptation, and Personalization*, 2011.

[27] D. He, Z. Peng, L. Hong and Y. Zhang, "A social reputation management for web communities," in *International Conference on Web-Age Information Management*, 2011.

[28] A. Gutowska and A. Sloane, "Evaluation of Reputation Metric for the B2C e-Commerce Reputation System," in *WEBIST*, 2009.

[29] W. Edwards, "How to use multiattribute utility measurement for social decisionmaking," *IEEE transactions on systems, man, and cybernetics,* vol. 7, no. 5, pp. 326-340, 1977.

[30] W. Edwards and F. H. Barron, "SMARTS and SMARTER: Improved simple methods for multiattribute utility measurement," *Organizational Behavior & Human Decision Processes,* vol. 60, pp. 306-325, 1994.

[31] U. Kumar, A. Ahmadi, A. K. Verma and P. e. Varde, Current Trends in Reliability, Availability, Maintainability and Safety: An Industry Perspective, Springer, 2015.

[32] P. Saripalli and G. Pingal, "Madmac: Multiple attribute decision methodology for adoption of clouds," in *2011 IEEE International Conference on Cloud Computing (CLOUD)*, 2011.

[33] P. K. Yoon and C.-L. Hwang, Multiple attribute decision making: an introduction, vol. 104, Sage publications, 1995.

[34] Adriyendi, "Multi-Attribute Decision Making Using Simple Additive Weighting and Weighted Product in Food Choice," *International Journal of Information Engineering and Electronic Business,* vol. 6, pp. 8-14, 2015.

[35] P. W. Bridgman, Dimensional Analysis, Yale University Press, 1922.

[36] E. Triantaphyllou, Multi-criteria decision making methods." Multi-criteria Decision Making Methods: A Comparative Study, Springer US, 2000.

[37] C.-L. Hwang and K. Yoon, Multiple attribute decision making: methods and applications a state-of-the-art survey, vol. 186, Springer Science & Business Media, 2012.