Robust Multi-Dimensional Trust Computing Mechanism for Cloud Computing

Mohamed Firdhous[1], Osman Ghazali[2], Suhaidi Hassan[2]
[1]*Faculty of Information Technology, University of Moratuwa, Moratuwa 10400, Sri Lanka.*
[2]*School of Computing, College of Arts and Sciences, Universiti Utara Malaysia, UUM Sintok 10600, Kedah, Malaysia.*
osman@uum.edu.my

***Abstract*—Cloud computing has become the most promising way of purchasing computing resources over the Internet. The main advantage of cloud computing is its economic advantages over the traditional computing resource provisioning. For cloud computing to become acceptable to wider audience, it is necessary to maintain the QoS commitments specified in the service level agreement. In this paper, the authors propose robust multi-level trust computing mechanism that can be used track the performance of cloud systems using multiple QoS attributes. Tests carried out show that the proposed mechanism is more robust than the ones published in the literature.**

***Index Terms*—Cloud Computing; Quality of Service; Trust Computing.**

I. INTRODUCTION

Electricity, water, gas and telephony are commonly known as utilities where the users are totally isolated from the nitty-gritty of the production process and pay only for the services they consume. Similarly cloud computing also makes the computing resources including infrastructure, development environment and applications available over the Internet and requires them to pay for the resources accessed. This has earned cloud computing the nick name 5th utility [1].

Cloud systems have been hosted as virtual system on top of the physical hardware [2]. Thus hardware virtualization is the enabling technology for cloud computing. The virtual systems thus hosted The virtual machine manager installed on the bare metal hardware divides the physical hardware into multiple computing units either using the time division technology, space division or combination of both [3]. The space division virtualization technology assigns dedicated hardware such as CPU cores, memory, I/O devices etc., to various processes, when available. On the other hand, time division virtualization technology divides all the hardware into multiple time slots and assigns them to different processes on a time shared basis [4]. These virtualized systems can be brought up and removed on demand [2]. Cloud computing services such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) are hosted on top of the virtualized systems as shown in Figure 1.

Due to its efficiency and profitability, cloud computing has attracted many service providers [5]. These service providers host their services and make them available over the Internet for customers to access. The quality of services provided by these providers would heavily depend on the capacity of the physical resources and the number of clients accessing them concurrently. At the commencement of services, the service providers and the clients enter into a Service Level Agreement (SLA) that specifies conditions and commitments to be satisfied by both parties [6]. In these agreements, the Quality of Service (QoS) to be satisfied by the provider would occupy an important place [7]. Thus the quality of service of the service providers would play an important role in identifying the right service provider. QoS is characterized generally with the attributes such as response time, delay, service time and preferred values for these attributes. Also, the dynamic nature of cloud computing requires continuous monitoring of these attributes [6].

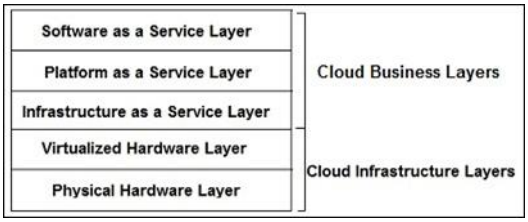

Figure 1: Cloud Computing Layered Model

This paper has been divided into five main sections as follows: Section 1 provides the introduction and background information on the issues handled in the paper and the proposed solution. Section 2 critically analyzes the trust computing mechanisms proposed in the literature with special reference to their shortcomings. Section 3 introduces the proposed robust multi-dimensional trust computing mechanism for cloud computing. Section 4 describes the experimental setup used for testing the proposed mechanism along with an in depth analysis on the results. Finally, Section 5 concludes the paper summarizing the findings with reference to the objectives set in Section 1.

II. RELATED WORK

This section takes an in depth look at the related work carried out by other researchers and published in journals, conference proceedings and technical reports. A critical analysis is carried out on two main areas of interest. They are namely, quality of service in cloud computing and trust computing in distributed systems.

Due to the similarity and multi-faceted nature of trust and service quality, trust computing mechanisms can be used to quantify the QoS of cloud systems [8]. Several trust computing mechanisms based on different criteria and functions have been reported in the literature [9-15]. Though, these mechanisms are based on strong algorithms and functions, they mainly suffer from that shortcoming that they take only one input attribute for computing the trust score.

Thus, the multi-faceted nature of trust as well as the user requirements for quantifying QoS on multiple attributes is totally ignored by these mechanisms. Hence the practical use of these mechanisms in a business cloud system is limited. In order to fill this shortcoming, the authors propose a multi-dimensional trust computing mechanism that incorporates statistical verification and non-linear hysteresis function. The robustness of the mechanism is enhanced by the statistical verification of the inputs and the non-linear hysteresis function in the events of short term temporary fluctuations and malicious attacks on the system [13, 14].

III. Robust Multi-Dimensional Trust Computing Mechanism

Trust computing mechanism mainly concentrates on trust evolution where the trust scores are either improved or worsened based on the results of the interactions [16]. Figure 2 shows the block diagram of the trust computing system proposed in the paper. The trust computing unit and the QoS monitoring unit make the trust computing system. The cloud provider is external to the system, but provides the actual QoS information after every interaction.

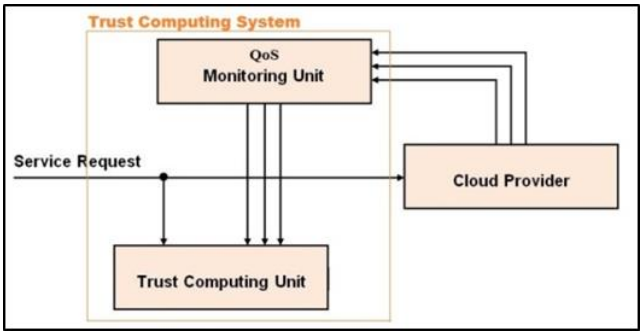

Figure 2: Trust Computing System

When a client signs a service level agreement with a service provider, he or she also signs up with a trust provider who is independent of both the service provider and the client. The client provides the trust provider with a committed QoS values along with the weights and confidence level for each attribute depending on the stringency of the service quality required. When the client request reaches the service provider, it is also given to the trust computing system. The trust computing system, then extracts the expected QoS parameters and expected values (specified in the SLA) from its database for the particular request. When the service is completed, the QoS monitoring units follows the actual performance values and supplies them to the trust computing unit.

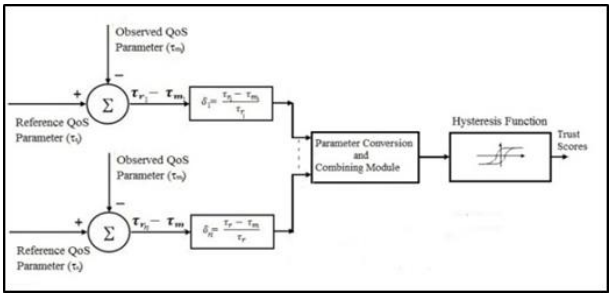

Figure 3: Trust Computing Unit

Figure 3 shows the trust computing unit in detail. The summer computes the difference between the actual value and the expected value for every attribute and supplies those differences to the next stage for computing the normalized attribute value. The normalization process removes any skewness in results due to the domination of a single attribute over the others. The parameter conversion and combining unit creates a single value by combining all the input parameters into a single value that can be supplied to the hysteresis function for computing the trust score.

The parameter conversion and combination is one of the main components of this mechanism that makes it multi-dimensional as opposed to all the other mechanisms. All the input parameters are converted to a single (combined) parameter as follows in Equation 1.

$$\tau = \frac{\alpha_1\tau_1 + \alpha_2\tau_2 + \cdots + \alpha_n}{\alpha_{1\alpha} + \alpha_2 + \cdots + \alpha_n} \qquad (1)$$

$$\alpha_1 + \alpha_2 + \cdots + \alpha_n = 1$$

where τ_r is the r[th] parameter and $\alpha_r$is the weight applied to it respectively.

The weights are selected depending on the importance of the parameter for the performance of the application. When an attribute does not play any role in the performance, its weight would be made equal to zero which essentially eliminates it from the trust computation process. Once the actual performance values (τ_a) are received, they are stored in the temporary storage for the purpose of computing the confidence interval. If the performance of any attribute falls within the confidence interval, the system performance is taken as satisfactory and eliminated from the computation of trust by making its weight (α) equal to zero. Figure 4 shows the trust computing algorithm employed in this mechanism.

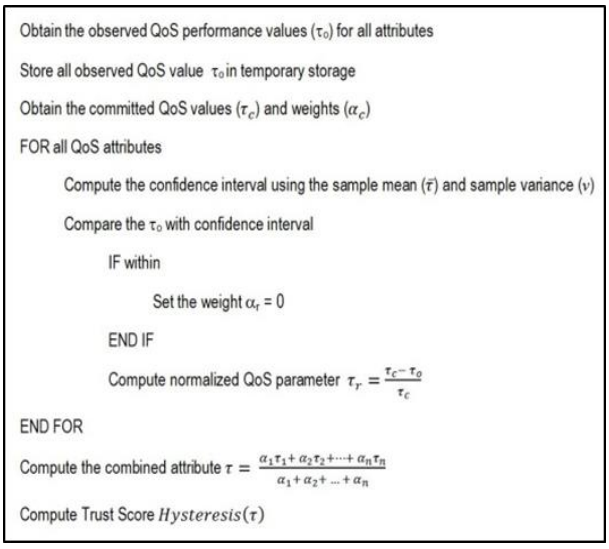

Figure 4: Trust Computing Algorithm

IV. Result and Discussion

The proposed mechanism was its functionality and accuracy with simulations. The simulation environment was created with Mat lab by creating every functional unit, independently and combining them together to form the

complete system. The hysteresis function in the trust computing unit was constructed as follows:

$$hysteresis\ (x) = \begin{cases} sigm\ (x-k), & for\ x_n > x_{n-1} \\ sigm\ (x+k), & for\ x_n < x_{n-1} \end{cases} \quad (2)$$

where k is the horizontal shift and $sigm(x) = \frac{1-e^{-x}}{1-e^{+x}}$. Sigm(x) is known as the sigmoid function that has an odd symmetry about the y-axis. The hysteresis loop thus created is shown in Figure 5.

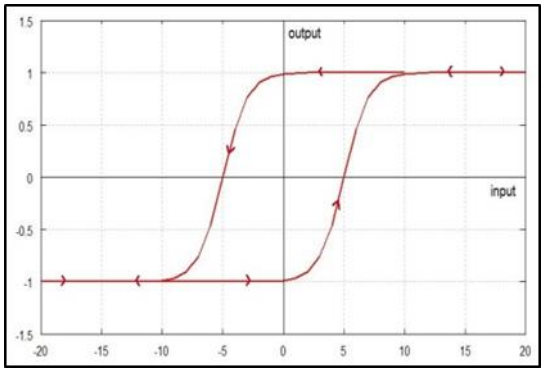

Figure 5: Hysteresis loop

Figure 6 shows the trust scores computed using two attributes along with the effect of weights applied on the input parameters. From Figure 6, it can be seen that the final trust score is more aligned towards the parameter that is applied a higher weight.

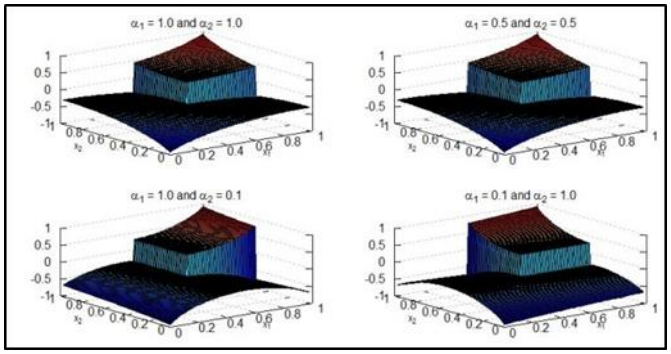

Figure 6: Effect of Multiple Attributes on Trust Scores

Figure 7 shows the trust values computed using the proposed mechanism along with that of the entropy based mechanism. The proposed mechanism was also tested using statistically validated inputs and non-validated inputs. The statistically validation checks if the change in the attribute is due to a temporary fluctuation or due to system degradation. If the observed input value falls within the confidence interval, it was taken as a temporary fluctuation and the effect of the attribute on the trust score was eliminated by making the weight (α) equal to zero. This way, if all the QoS attributes fall within their respective confidence intervals, then the trust score will not be modified from the previous value as there is no observable change in performance. From Figure 7, it can be seen that the performance of the proposed mechanism is better and subject to less fluctuations compared to the entropy based mechanism proposed by Dai et al [11]. Also it could be seen that when the statistically validated input is applied to the proposed mechanism it shows more robust performance as small fluctuations in the performance is suppressed by the statistical validation process.

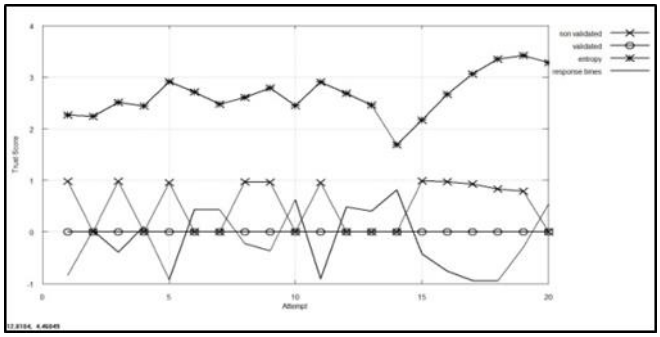

Figure 7: Comparison of Trust Scores Computed

Figure 8 shows the effect of the confidence level on the trust scores computed. From this figure, it could be seen that the trust scores computed using 90% confidence level shows more fluctuations than the one computed using 95% confidence level. This is due to the reason that at 95% confidence level, the expectation of the client on performance is more stringent.

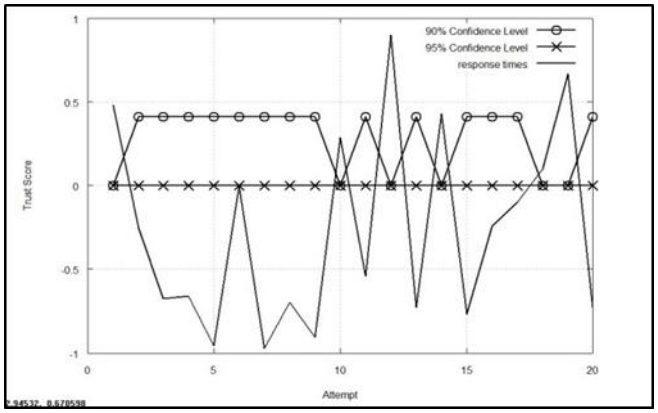

Figure 8: Effect of Confidence Level on Trust Scores

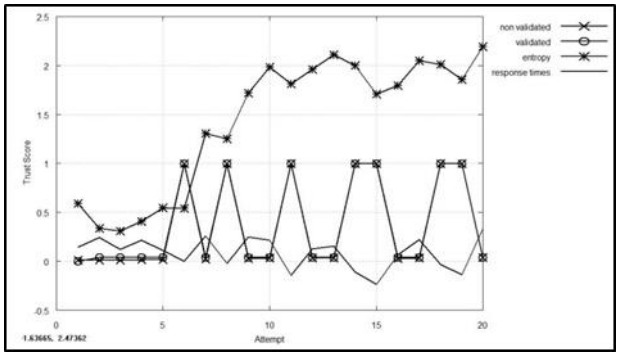

Figure 9: Effect of Large Fluctuations on Trust Scores

Hence it can be concluded that the proposed mechanism performs better and more robust than the entropy based mechanism in the events of temporary fluctuations. Also it cannot be attacked by adversaries by continuous bombardments. Figure 9 shows trust scores computed using the same methods when the fluctuations are large. From Figure 9, it can be seen that when the fluctuations are large trust scores show the same performance for both validated and non-validated inputs. This is due to the reason that when

the fluctuations are large, they are due to actual system degradation than temporary ones.

V. CONCLUSION

In this paper, the authors presented a robust multi-dimensional trust computing mechanism that can track the performance of a cloud system using more than on QoS parameter. The mechanisms proposed in the literature so far are all single dimension as they compute the trust score using only one input parameter. More over the proposed mechanism shows more robust performance than the ones that are implemented using monotonously changing functions. When the proposed mechanism is equipped with additional statistical validation of inputs, its performance becomes better due to double protection provided by statistical validation and hysteresis loop both are immune to small changes in inputs.

ACKNOWLEDGMENT

This research was supported in part by the Fundamental Research Grant Scheme (FRGS) of the Ministry of Higher Education of the Government of Malaysia under Grant No. 13144.

REFERENCES

[1] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, I. Brandic. Cloud computing and emerging IT platforms: Vision, hype and reality for delivering computing as the 5th utility, Future Generation Computer Systems, 25(6) (2009) 599-616.

[2] B. Siddhisena, L. Warusawithana, M. Mendis. Next generation multi-tenant virtualization cloud computing platform. 13th International Conference on Advanced Communication Technology, Seoul, South Korea, (2011) 405-410.

[3] A. A. Semnanian, J. Pham, B. Englert, X. Wu. Virtualization technology and its impact on computer hardware architecture. 8th International Conference on New Generation Information Technology, Las Vegas, NV, USA, (2011) 719-724.

[4] S. Zaman, D. Grosu. Combinatorial auction-based allocation of virtual machine instances in clouds. 2nd IEEE International Conference on Cloud Computing Technology and Science, Indianapolis, IN, USA, (2010) 127-134.

[5] B. P. Rimal, E. Choi, I. Lumb. A taxonomy and survey of cloud computing systems. 5th International Joint Conference on INC, IMS and IDC, Seoul, Korea, (2009) 44-51.

[6] P. Patel, A. Ranabahu, A. Sheth. Service level agreement in cloud computing. ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications, Orlando, FL, USA, (2009) 1-10.

[7] L. Wu, R. Buyya. Service level agreement in utility computing systems. In V. Cardellini, E. Casalicchio, K. C. Branco, J. Estrella, F. Monaco (Eds.), Performance and Dependability in Service Computing: Concepts, Techniques and Research Directions, Information Science Reference, Hershey, PA, USA, (2012) 1-25.

[8] M. Firdhous, O. Ghazali, S. Hassan, N. Z. Harun, A. Abas. Honey bee based trust management system for cloud computing. 3rd International Conference on Computing and Informatics (ICOCI 2011), Bandung, Indonesia, (2011) 327–332.

[9] H. Chen, Z. Ye. Research of P2P trust based on fuzzy decision making. 12th International Conference on Computer Supported Cooperative Work in Design, Xi'an, China, (2008) 793-796.

[10] CQ. Tian, SH. Zou, WD. Wang, SD. Cheng. A new trust model based on recommendation evidence for P2P networks. Chinese Journal of Computers, 31(2) (2008) 270-281.

[11] H. Dai, Z. Jia, X. Dong. An entropy-based trust modeling and evaluation for wireless sensor networks. International Conference on Embedded Software and Systems, Chengdu, Sichuan, China, (2008) 27-34.

[12] M. Firdhous, O. Ghazali, S. Hassan. A trust computing mechanism for cloud computing. 4th ITU Kaleidoscope Academic Conference, Cape Town, South Africa, (2011) 199-205.

[13] M. Firdhous, O. Ghazali, S. Hassan. A trust computing mechanism for cloud computing with multilevel thresholding. 6th International Conference on Industrial & Information Systems (ICIIS2011), (2011) 457-461; Kandy, Sri Lanka.

[14] M. Firdhous, O. Ghazali, S. Hassan. Hysteresis-based robust trust computing mechanism for cloud computing, IEEE Region 10 Conference (TENCON 2012), (2012) 796-801; Cebu, the Philippines.

[15] M. Firdhous, O. Ghazali, S. Hassan. A memoryless trust computing mechanism for cloud computing. 4th International Conference on Networked Digital Technologies (NDT'2012), Dubai, (2012) 174-185.

[16] A. S. Abari, T. White. DART: A distributed analysis of reputation and trust framework. Computational Intelligence, 28(4) (2012) 642-682.