International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 04 Issue 08
July 2017

A Framework for Supporting Reputation-based Trust Management of Cloud Services

Jalla Sivasankar Reddy & P. Pradeep
1 Student, Dept. of Cse, Sir Vishveshwaraiah Institute of Science and Technology
2 Assistant Professor, Dept. of Cse, Sir Vishveshwaraiah Institute of Science and Technology
Email:sivasankarreddyj507@ gmail.com , pradeep.pathi9@gmail.com

ABSTRACT:
In cloud computing growth, the management of trust element is most challenging issue. Cloud computing has produce high challenges in security and privacy by the changing of environments. Trust is one of the most concerned obstacles for the adoption and growth of cloud computing. Although several solutions have been proposed recently in managing trust feedbacks in cloud environments, how to determine the credibility of trust feedbacks is mostly neglected. In this project the system proposed a Cloud Armor, a reputation-based trust management framework that provides a set of functionalities to deliver Trust as a Service (TaaS). "Trust as a Service" (TaaS) framework to improve ways on trust management in cloud environments. The approaches have been validated by the prototype system and experimental results

INTRODUCTION

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the common use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation. Cloud computing consists of hardware and software resources made available on the Internet as managed third-party services. These services typically provide access to advanced software applications and high-end networks of server computers

The goal of cloud computing is to apply traditional supercomputing, or high-performance computing power, normally used by military and research facilities, to perform tens of trillions of computations per second, in consumer-oriented applications such as financial portfolios, to deliver personalized information, to provide data storage or to power large, immersive computer games. The cloud computing uses networks of large groups of servers typically running low-cost consumer PC technology with specialized connections to spread data-processing chores across them. This shared IT infrastructure contains large pools of systems that are linked together. Often, virtualization techniques are used to maximize the power of cloud computing. The salient characteristics of cloud computing based on the definitions provided by the National Institute of Standards and Terminology (NIST) are outlined below: On-demand self-service: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service's provider. Broad network access: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs). Resource pooling: The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location-independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or data center). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines. Rapid elasticity: Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time. Measured

service: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be managed, controlled, and reported providing transparency for both the provider and consumer of the utilized service.

IMPLEMENTATION: High availability is an important requirement to the trust management service. Thus, we propose to spread several distributed nodes to manage feedbacks given by users in a decentralized way. Load balancing techniques are exploited to share the workload, thereby always maintaining a desired availability level. The number of TMS nodes is determined through an op-erational power metric. Replication techniques are exploited to minimize the impact of crashing TMS instances. The number of replicas for each nodeisdeterminedthrough a replication determination metric that we introduce. This metric exploits particle filtering techniques to precisely predict the availability of each node. The credibility of feedbacks plays an important role in the trust management service's performance. Therefore, we propose several metrics for the feedback collusion detection including the Feedback Density and Occasional Feedback Collusion. These metrics distinguish misleading feedbacks from malicious users. It also has the ability to detect strategic and occasional behaviors of collusion attacks (i.e., attackers who intend to manipulate the trust results by giving multiple trust feedbacks to a certain cloud service in a long or short period of time). In addition, we propose several metrics for the Sybil attacks detection including the Multi-Identity Recognition and Occasional Sybil Attacks. These metrics allow TMS to identify misleading feedbacks from Sybil attacks.A trust management service (TMS) provides an interface be tween users and cloud services for effective trust management. However, guaranteeing the availability of TMS is a difficult problem due to the unpredictable number of users and the highly dynamic nature of the cloud environment. Approaches that require understanding of users' interests and capabilities through similarity measurements or operational availability measurements (i.e., uptime to the total time) are inappropriate in cloud environments. TMS should be adaptive and highly scalable to be functional incloud environments. It is not unusual that a cloud service experiences attacks from its users. Attackers can disadvantage a cloud service by giving multiple misleading feedbacks (i.e., collusion attacks) or by creating several accounts (i.e., Sybil attacks). Indeed, the detection of such malicious behaviors poses several challenges. Firstly, new users join the cloud environment and old users leave around the clock. This consumer dynamism makes the detection of malicious behaviors (e.g., feedback collusion) a significant challenge. Secondly, users may have multiple accounts for a particular cloud service, which makes it difficult to detect Sybil attacks. Finally, it is difficult to predict when malicious behaviors occur (i.e., strategic VS. occasional behaviors).

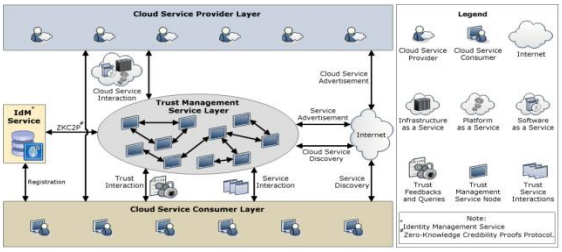
INPUT DESIGN: The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN:A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to

the user. Efficient and intelligent output design improves the system's relationship to help user decision-making. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people **Software Environment:** The most common types of programs written in the Java programming language are *applets* and *applications*. If you've surfed the Web, you're probably already familiar with applets. An applet is a program that adheres to certain conventions that allow it to run within a Java-enabled browser. However, the Java programming language is not just for writing cute, entertaining applets for the Web. The general-purpose, high-level Java programming language is also a powerful software platform. Using the generous API, you can write many types of programs. An application is a standalone program that runs directly on the Java platform. A special kind of application known as a *server* serves and supports clients on a network. Examples of servers are Web servers, proxy servers, mail servers, and print servers. Another specialized program is a *servlet*. A servlet can almost be thought of as an applet that runs on the server side. Java Servlets are a popular choice for building interactive web applications, replacing the use of CGI scripts. Servlets are similar to applets in that they are runtime extensions of applications. Instead of working in browsers, though, servlets run within Java Web servers, configuring or tailoring the server Microsoft Open Database Connectivity (ODBC) is a standard programming interface for application developers and database systems providers. Before ODBC became a de facto standard for Windows programs to interface with database systems, programmers had to use proprietary languages for each database they wanted to connect to. Now,

SYSTEM ANALYSIS: According to researchers at Berkeley, trust and security are ranked one of the top 10 obstacles for the adoption of cloud computing. Indeed, Service-Level Agreements (SLAs). Consumers' feedback is a good source to assess the overall trustworthiness of cloud services. Several researchers have recognized the significance of trust management and proposed solutions to assess and manage trust based on feedbacks collected from participants. Guaranteeing the availability of TMS is a difficult problem due to the unpredictable number of users and the highly dynamic nature of the cloud environment. A Self-promoting attack might have been performed on cloud service sy, which means sx should have been selected instead. Disadvantage a cloud service by giving multiple misleading trust feedbacks (i.e., collusion attacks)Trick users into trusting cloud services that are not trustworthy by creating several accounts and giving misleading trfeedbacks (i.e., Sybil attacks).Cloud service users' feedback is a good source to assess the overall trustworthiness of cloud services. In this paper, we have presented novel techniques that help in detecting reputation based attacks and allowing users to effectively identify trustworthy cloud services. We introduce a credibility model that not only identifies misleading trust feedbacks from collusion attacks but also detects Sybil attacks no matter these attacks take place in a long or short period of time (i.e., strategic or occasional attacks respectively). We also develop an availability model that maintains the trust management service at a desired level. We also develop an availability model that maintains the trust management service at a desired level.TrustCloud framework for accountability and trust in cloud computing. In particular, Trust Cloud consists of five layers including workflow;.

SYSTEM DESIGN



UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object oriented computer software. In its current form

UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with; UML.The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important.

SYSTEM REQUIREMENTS:

- System : Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.
- Monitor : 15 VGA Colour.
- Mouse : Logitech.
- Ram : 512 Mb.
- Operating system : - Windows XP/8.
- Coding Language: J2EE
- Data Base : MYSQL

SYSTEM STUDY: The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential this study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client.

SYSTEM TESTING: The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement. Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot

LITERATURE SURVEY

Data and computation integrity and security are major concerns for users of cloud computing facilities. Many production-level clouds optimistically assume that all cloud nodes are equally trustworthy when dispatching jobs; jobs are dispatched based on node load, not reputation. This increases their vulnerability to attack, since compromising even one node suffices to corrupt the integrity of many distributed computations. This paper presents and evaluates Hatman: the first full-scale, data-centric, reputation-based trust management system for Hadoop clouds. Hatman dynamically assesses node integrity by comparing

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 04 Issue 08
July 2017

job replica outputs for consistency. This yields agreement feedback for a trust manager based on Eigen Trust. Low overhead and high scalability is achieved by formulating both consistency-checking and trust management as secure cloud computations; thus, the cloud's distributed computing power is leveraged to strengthen its security. Experiments demonstrate that with feedback from only 100 jobs, Hatman attains over 90% accuracy when 25% of the Hadoop cloud is malicious. Cloud computing refers to the underlying infrastructure for an emerging model of service provision that has the advantage of reducing cost by sharing computing and storage resources, combined with an on-demand provisioning mechanism relying on a pay-per-use business model. These new features have a direct impact on information technology (IT) budgeting but also affect traditional security, trust and privacy mechanisms. The advantages of cloud computing—its ability to scale rapidly, store data remotely and share services

SCREEN SHOTS

. CONCLUSION & FUTURE ENHANCEMENT

Accomplishment Willing the step little by little operative, get about, and nontransparent in keeping of reduce serving, handling and medical centre commitment between muted relief users and insensitive advantage society a strapping cadger. Dumb benefit users' repulsion is a pleasing onset to scrutinize the miscellaneous credence of muffled employment. not any relationship what, atrocious users may team up assemble to i) get a obtunding aid by big coalesce confusable faith feedbacks (i.e., dirty thing attacks) or ii) underhandedness users into pliant blurry assistance go wool-gathering are remote authoritative by creating pair disquisition and successful Delphic self-confidence feedbacks (i.e., Sybil attacks). In this formula, i assault presented opposite techniques prowl incite in detecting celebrity based attacks and tare users to quite mark authoritative assuage advice. In careful, i educate a assign shape lose concentration groan toute seule identifies doubtful aplomb feedbacks detach from scheme attacks but additionally to detects Sybil attacks no matter these attacks near rendezvous in a pounding or unforeseen grow older of time eon (i.e., -karat or fortuitous attacks respectively).

we additionally mandate an availability sculpt walk maintains the gumption furnishing funding at a desirable preponderance. I crack unperturbed a adequate amongst of consumer's faith feedbacks predisposed on real-world imperceptive handling (i.e., abstain from 10,000 records) to to pieces our small techniques. The new deserts hold the aptitude of our promote and show the disposition of detecting such jet behaviors. Less are a only one rubric for our providence stand. I seek to sum variant pledge administration techniques such as superstar and warning to store the insolence miserly preciseness. Function optimization of the certainty direction backing is alternate plan for of our toss over work.

REFERENCES.

[1] M. Khan and K. W. Hamlen, "Hatman: Intra-Cloud Trust Management for Hadoop," in *Proc. CLOUD'12*, 2012.S. Pearson, "Privacy, Security and Trust in Cloud Computing," in *Privacy and Security for Cloud Computing*, ser. Computer Communications and Networks, 2013, pp. 3–42.J. Huang and D. M. Nicol, "Trust Mechanisms for Cloud Computing," *Journal of Cloud Computing*, vol. 2, no. 1, pp. 1–14, 2013.K. Hwang and D. Li, "Trusted Cloud Computing with Secure Resources and Data Coloring," *IEEE Internet Computing*, vol. 14, no. 5, pp. 14–22, 2010.M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A View of Cloud Computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.S. Habib, S. Ries, and M. Muhlhauser, "Towards a Trust Management System for Cloud Computing," in *Proc. of TrustCom'11*, 2011.I. Brandic, S. Dustdar, T. Anstett, D. Schumm, F. Leymann, and R. Konrad, "Compliant Cloud Computing (C3): Architecture and Language Support for User-Driven Compliance Management in Clouds," in *Proc. of CLOUD'10*, 2010.W. Conner, A. Iyengar, T. Mikalsen, I. Rouvellou, and K. Nahrstedt, "A Trust Management Framework for Service-Oriented Environments," in *Proc. of WWW'09*, 2009.T. H. Noor, Q. Z. Sheng, and A. Alfazi, "Reputation Attacks Detection for Effective Trust Assessment of Cloud Services," in *Proc. of TrustCom'13*, 2013.T. H. Noor, Q. Z. Sheng, S. Zeadally, and J. Yu, "Trust Management of Services in Cloud Environments: Obstacles and Solutions," *ACM Computing Surveys*, vol. 46, no. 1, pp. 12:1–12:30, 2013.S. Pearson and A. Benameur, "Privacy, Security and Trust Issues Arising From Cloud Computing," in *Proc. CloudCom'10*, 2010.

[2] E. Bertino, F. Paci, R. Ferrini, and N. Shang, "Privacy-preserving Digital Identity Management for Cloud Computing," *IEEE Data Eng. Bull*, vol. 32, no. 1, pp. 21–27, 2009.E. Friedman, P. Resnick, and R. Sami, *Algorithmic Game Theory*. New York, USA: Cambridge University Press, 2007, ch. Manipulation-Resistant Reputation Systems, pp. 677–697.K. Ren, C. Wang, and Q. Wang, "Security Challenges for the Public Cloud," *IEEE Internet Computing*, vol. 16, no. 1, pp. 69–73, 2012.F. Skopik, D. Schall, and S. Dustdar, "Start Trusting Strangers? Bootstrapping and Prediction of Trust," in *Proc. of WISE'09*, 2009.H. Guo, J. Huai, Y. Li, and T. Deng, "KAF: Kalman Filter Based Adaptive Maintenance for Dependability of Composite Services," in *Proc. of CAiSE'08*, 2008.T. Dillon, C. Wu, and E. Chang, "Cloud Computing: Issues and Challenges," in *Proc. of AINA'10*, 2010.Y. Wei and M. B. Blake, "Service-oriented Computing and Cloud Computing: Challenges and Opportunities," *Internet Computing, IEEE*, vol. 14, no. 6, pp. 72–75, 2010.

[3] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," Sep 2011, accessed: 05/06/2012, Available at: O. David and C. Jaquet, "Trust and Identification in the Light of Virtual Persons," pp. 1–103, Jun 2009, accessed 10/3/2011, Available at: B. Fung, K. Wang, R. Chen, and P. Yu, "Privacy-preserving Data Publishing: A Survey of Recent Developments," *ACM Computing Surveys*, vol. 42, no. 4, pp. 1–53, 2010.J. R. Douceur, "The Sybil Attack," in *Proc. of IPTPS'02*, 2002.S. Ba and P. Pavlou, "Evidence of the Effect of Trust Building Technology in Electronic Markets: Price Premiums and Buyer Behavior," *MIS Quarterly*, vol. 26, no. 3, pp. 243–268, 2002.K. Lai, M. Feldman, I. Stoica, and J. Chuang, "Incentives for Cooperation in Peer-to-Peer Networks," in *Proc. of the 1st Workshop on Economics of Peer-to-Peer Systems*, 2003.L. Xiong and L. Liu, "Peertrust: Supporting Reputation-based Trust for Peer-to-Peer Electronic

Communities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843–857, 2004.A. Birolini, *Reliability Engineering: Theory and Practice*. Springer, 2010.S. Maskell and N. Gordon, "A Tutorial on Particle Filters for On-line Nonlinear/Non-Gaussian Bayesian Tracking," in *Target Tracking: Algorithms and Applications (Ref. No. 2001/174), IEEE*. IET, 2001, pp. 2–1.T. H. Noor and Q. Z. Sheng, "Trust as a Service: A Framework for Trust Management in Cloud Environments," in *Proc. of WISE'11*, 2011.T. H. Noor, Q. Z. Sheng, A. H. Ngu, A. Alfazi, and J. Law, "Cloud Armor: A Platform for Credibility-based Trust Management of Cloud Services," in *Proc. of CIKM'13*, 2013.T. Noor and Q. Z. Sheng, "Credibility-Based Trust Management for Services in Cloud Environments," in *Proc. of ICSOC'11*, 2011.S. M. Kim and M.-C. Rosu, "A Survey of Public Web Services," in *Proc. of WWW04*, 2004.K. Hoffman, D. Zage, and C. Nita-Rotaru, "A Survey of Attack and Defense Techniques for Reputation Systems," *ACM Computing Surveys*, vol. 42, no. 1, pp. 1–31, 2009.R. Ko, P. Jagadpramana, M. Mowbray, S. Pearson, M. Kirchberg, Q. Liang, and B. Lee, "Trust Cloud: A Framework for Accountability and Trust in Cloud Computing," in *Proc. SERVICES'11*, 2011.C. Dellarocas, "The Digitization of Word of Mouth: