Cloud Service Efficiency Analysis Based on Trust Evaluation Using Secure Feedback System

Abdhul Rahuma H[1], Ajay Karthik R[2], Ejas Ahmed M[3], Mushthak Ahamed S[4]
[1]Assistant professor Dept of Information Technology
[2]Dept of Information Technology
[1, 2] Saranathan College of Engineering, Trichy

Abstract- *Cloud provides many services to the cloud users including the data storage and data sharing process. The trustworthiness of cloud services is an important consideration for making cloud selection decision. Here implement a distributed trust evaluation protocol with privacy protection system. This protocol allows a cloud service provider to certify a user eligibility to post feedback. It allows the filtering of extreme ratings without leaking privacy information. Encryption methodology used to encrypt the users data and feedback information.*

I. INTRODUCTION

A distributed trust evaluation protocol with privacy protection system is introduced for Intercloud management. This has following parties cloud service provider (CSP); user/rater; enquirer; distributed feedback storage (FBS) and secret sharing network. Under the Intercloud system model, the CSPs provide cloud services collaboratively to users, and serve each other as well. In general, there are two types of cloud service users: consumer users and business users. Consumer users use a cloud service. They have only a one-way trust/service relationship with a CSP (i.e., the service is one-way CSPs serving consumer users). Business users may provide services to each other (i.e., two-way trust/service relationship). After using a cloud service or under certain arrangements, the users can rate the service or a trust attribute (e.g., availability, response time, price, technical support). That means the users are also raters during the feedback/rating process. Note that to facilitate the explanation, we focus on evaluating one service or trust attribute. It can easily be extended to evaluate multiple services or trust attributes. For the Intercloud system, the FBS can be implemented by the Intercloud exchanges, or by the cloud service providers themselves using a blockchain-based system. Alternatively, other distributed storage systems, with integrity guarantees, can also be used to store feedback. We also assume there is a secret sharing network for protecting encrypted feedback. Detailed operations will be explained later. The secret sharing network can be formed by the users/raters to protect their feedback privacy. Alternatively, it can be provided by a third party as a service. The secret sharing mechanism is conducted in a distributed manner, without direct interaction between raters.

II. CLOUD COMPUTING

Cloud computing is a computing paradigm, where a large pool of systems are connected in private or public networks, to provide dynamically scalable infrastructure for application, data and file storage. With the advent of this technology, the cost of computation, application hosting, content storage and delivery is reduced significantly. It is a practical approach to experience direct cost benefits and it has the potential to transform a data center from a capital-intensive set up to a variable priced environment. The idea of cloud computing is based on a very fundamental principles of reusability of IT capabilities. The difference that cloud computing brings compared to traditional concepts of “grid computing”, “distributed computing”, “utility computing”, or “autonomic computing” is to broaden horizons across organizational boundaries. Forrester [1] defines cloud computing as: “A pool of abstracted, highly scalable, and managed compute infrastructure capable of hosting end customer applications and billed by consumption”. It is a technology that uses the internet and central remote servers to maintain data and applications and allows consumers and businesses to use applications without installation and access their personal files at any computer with internet access. This technology allows for much more efficient computing by centralizing data storage, processing and bandwidth. Cloud computing examples are Yahoo email, Gmail, or Hotmail.

2.1 SERVICE MODELS OF CLOUD

Cloud Providers offer services that can be grouped into three categories.

1. Software as a Service (SaaS)
2. Platform as a Service (Paas)
3. Infrastructure as a Service (Iaas)

Software as a Service

In this model, a complete application is offered to the customer, as a service on demand. A single instance of the service runs on the cloud & multiple end users are serviced. On the customers‟ side, there is no need for upfront investment in servers or software licenses, while for the provider, the costs are lowered, since only a single application needs to be hosted & maintained. Today SaaS is offered by companies such as Google, Salesforce, Microsoft, Zoho, etc.

Platform as a Service

Here, a layer of software, or development environment is encapsulated & offered as a service, upon which other higher levels of service can be built. The customer has the freedom to build his own applications, which run on the provider‟s infrastructure. To meet manageability and scalability requirements of the applications, PaaS providers offer a predefined combination of OS and application servers, such as LAMP platform (Linux, Apache, MySql and PHP), restricted J2EE, Ruby etc. Google‟s App Engine, Force.com, etc are some of the popular PaaS examples.

Infrastructure as a Service

IaaS provides basic storage and computing capabilities as standardized services over the network. Servers, storage systems, networking equipment, data centre space etc. are pooled and made available to handle workloads. The customer would typically deploy his own software on the infrastructure. Some common examples are Amazon, GoGrid, 3 Tera, etc.

2.2 DEPLOYMENT MODELS OF CLOUD

Enterprises can choose to deploy applications on Public, Private or Hybrid clouds. Cloud Integrators can play a vital part in determining the right cloud path for each organization.

Public Cloud

Public clouds are owned and operated by third parties; they deliver superior economies of scale to customers, as the infrastructure costs are spread among a mix of users, giving each individual client an attractive low-cost, “Pay-as-you-go” model. All customers share the same infrastructure pool with limited configuration, security protections, and availability variances. These are managed and supported by the cloud provider. One of the advantages of a Public cloud is that they may be larger than an enterprises cloud, thus providing the ability to scale seamlessly, on demand.

Private Cloud

Private clouds are built exclusively for a single enterprise. They aim to address concerns on data security and offer greater control, which is typically lacking in a public cloud. There are two variations to a private cloud:

1. On-premise Private Cloud
2. Externally hosted Private Cloud

On-premise Private Cloud

On-premise private clouds, also known as internal clouds are hosted within one’s own data center. This model provides a more standardized process and protection, but is limited in aspects of size and scalability. IT departments would also need to incur the capital and operational costs for the physical resources. This is best suited for applications which require complete control and configurability of the infrastructure and security.

Externally hosted Private Cloud

This type of private cloud is hosted externally with a cloud provider, where the provider facilitates an exclusive cloud environment with full guarantee of privacy. This is best suited for enterprises that don’t prefer a public cloud due to sharing of physical resources.

Hybrid Cloud

Hybrid Clouds combine both public and private cloud models. With a Hybrid Cloud, service providers can utilize 3rd party Cloud Providers in a full or partial manner thus increasing the flexibility of computing. The Hybrid cloud environment is capable of providing on-demand, externally provisioned scale. The ability to augment a private cloud with the resources of a public cloud can be used to manage any unexpected surges in workload.

2.3 CHALLENGES OF CLOUD COMPUTING

The following are some of the notable challenges associated with cloud computing, and although some of these may cause a slowdown when delivering more services in the cloud, most also can provide opportunities, if resolved with due care and attention in the planning stages.

Security and Privacy:

Perhaps two of the more “hot button” issues surrounding cloud computing relate to storing and securing

data, and monitoring the use of the cloud by the service providers. These issues are generally attributed to slowing the deployment of cloud services. These challenges can be addressed, for example, by storing the information internal to the organization, but allowing it to be used in the cloud. For this to occur, though, the security mechanisms between organization and the cloud need to be robust and a Hybrid cloud could support such a deployment.

Lack of Standards:

Clouds have documented interfaces; however, no standards are associated with these, and thus it is unlikely that most clouds will be interoperable. The Open Grid Forum is developing an Open Cloud Computing Interface to resolve this issue and the Open Cloud Consortium is working on cloud computing standards and practices. The findings of these groups will need to mature, but it is not known whether they will address the needs of the people deploying the services and the specific interfaces these services need. However, keeping up to date on the latest standards as they evolve will allow them to be leveraged, if applicable.

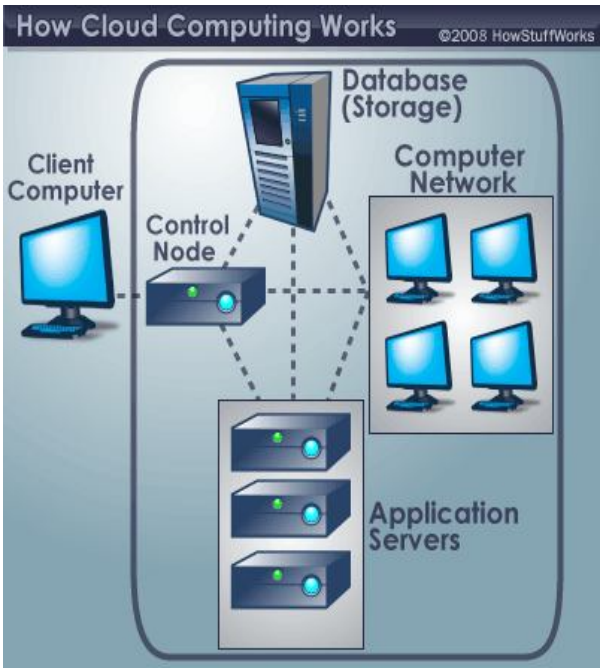
Continuously Evolving:

User requirements are continuously evolving, as are the requirements for interfaces, networking, and storage. This means that a "cloud," especially a public one, does not remain static and is also continuously evolving.

Compliance Concerns:

The Sarbanes-Oxley Act (SOX) in the US and Data Protection directives in the EU are just two among many compliance issues affecting cloud computing, based on the type of data and application for which the cloud is being used. The EU has a legislative backing for data protection across all member states, but in the US data protection is different and can vary from state to state. As with security and privacy mentioned previously, these typically result in Hybrid cloud deployment with one cloud storing the data internal to the organization.

2.4WORKING of cloud computing



Cloud computer works by hosting your information on computers 'out there' 'in the cloud'. The cloud is basically a batch of computers called data centers or servers that hold your information (files, images, videos, etc) and can be located anywhere. You're working in a cloud because you don't have to store software or files on your own computer.
To understand it works, it is important to think of cloud computing in two levels. The front level is the user level which is what you use such as your Google Email or Facebook or the webpage where you are downloading the latest version of Flash. The other level is the backend which is all of the hardware and the software architecture. It's here that the actual Gmail or Facebook is held waiting to be accessed by you.

Because all of the different servers are running together in cloud computing, one application can have the computer power of many servers. This allows something like Facebook to run. Imagine trying to host Facebook on your laptop. It just wouldn't work.Therefore, the cloud brings together a large number of computers to operate a single application. And anything that is stored anywhere other than your local hard drive—on your computer—is labeled as being in the clouds.

Advantages of Cloud Computing:

- No capital investment needed for technology infrastructure.
- Pay only for recourses that we use (utility computing).
- Immediate Access to large range of applications.

Disadvantages of Cloud Computing:

- Data Security - Cannot guarantee misuse of data at data centers.
- Data Theft – Hacking is on the increase and all data is exposed on the internet.
- Internet Connectivity - Low Speed or Downtime would impact productivity.
- Availability - No control over down time of Cloud Services.
- Sustainability - No control if Cloud Service Provider closes down.

III. EXISTING SYSTEM

Cloud service users' feedback is a good source to assess the overall trustworthiness of cloud services. However, malicious users may collaborate together to disadvantage a cloud service by giving multiple misleading trust feedbacks (i.e., collusion attacks) or trick users into trusting cloud services that are not trustworthy by creating several accounts and giving misleading trust feedbacks (i.e., Sybil attacks). In proposed work have presented a novel technique that help in detecting reputation based attacks and allowing users to effectively identify trustworthy cloud services. Introduce the concept of feedback density to support the determination of credible trust feedbacks. Specifically, we consider the total number of users who give trust feedbacks to a particular cloud service as the feedback mass, the total number of trust feedbacks given to the cloud service as the feedback volume. The feedback volume is influenced by the feedback volume collusion factor which is controlled by a specified volume collusion threshold. This factor regulates the multiple trust feedbacks extent that could collude the overall trusted feedback volume.

3.1 DISADVANTAGES

- Cloud service provider can easily remove negative comments about its services.
- Malicious users can generate a large volume of misleading feedback
- It supports only text feedback process.

IV. PROPOSED SYSTEM

Proposed method introduces a trust evaluation method that is used by the trust broker of Fig. 1 to evaluate the reliability of each customer feedback. In the proposed method, we focus on the reliability of the feedback of the middle customer 'B'. In the first step, we evaluate the reliability of a middle customer 'B' by comparing its feedbacks for commonly rated services. Each customer feedback is weighted based on a newly introduced similarity measure. In the second step, we consider all the feedbacks that customer 'B' has rated and compare them with the majority of feedbacks.

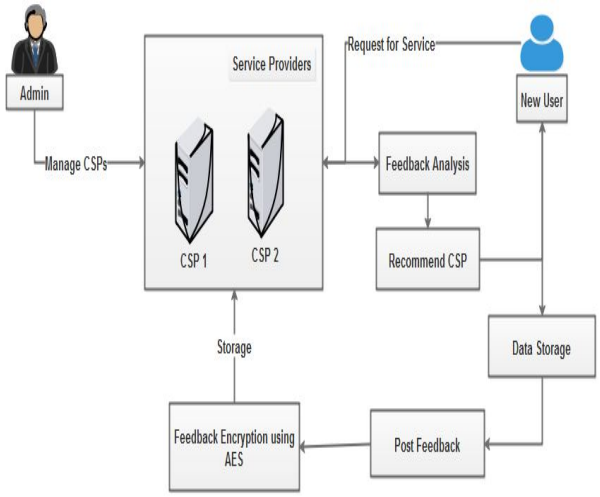
In this scheme, each new customer of a cloud service can determine its trust level using the past experiences of previous customers of this cloud service. The proposed method introduces an adaptive method that helps distinguish between malicious and reliable customer feedbacks. The proposed scheme assigns a reliability weight to each customer feedback. Users who have shared any cloud service before and did rate it similarly are more likely to have the same opinion of a new service. Hence, their evaluation of a new service should be given a higher weight. In cases where no customer has shared a common service with us before, then existing customers who have had a feedback closer to the majority (in any service) may seem to be more reliable and hence their opinion is given a higher weight. The balance between the feedback of customers with shared service and those of majority consensus in the final decision is determined dynamically and based on the rate of malicious attacks.

4.1 ADVANTAGES

- It allows the filtering of extreme ratings without leaking privacy information.
- Feedbacks are encrypted and stored securely.
- Fake reviews are avoided efficiently.

V. SYSTEM ARCHITECTURE

Software architecture involves the high level structure of software system abstraction, by using decomposition and composition, with architectural style and quality attributes. A software architecture design must conform to the major functionality and performance requirements of the system, as well as satisfy the non-functional requirements such as reliability, scalability, portability, and availability. Software architecture must describe its group of components, their connections, interactions among them and deployment configuration of all components.



Description

The proposed work has two users namely admin and user. Admin could create the framework which is designed to provide recommendation based cloud service selection process. Users should enroll and access cloud service, then provide the feedback about the service accessing from cloud environment. Registered users are only allowed to post feedbacks. Feedbacks are encrypted and stored on database securely. Then the system will analyse the user's feedbacks then predict, which service provider has highly recommended feedbacks. Then recommendation will be performed to shows the best service provider to the user.

5.1 MODULE

1. Cloud Framework
2. User Access
3. Review Encryption
4. Review Analysis
5. Recommendation Process

5.1.1 MODULE DESCRIPTION

5.1.1.1 CLOUD FRAMEWORK

The trustworthiness of cloud services is an important consideration for making cloud selection decision. Trust in a service is generally concerned with a belief in whether the service can be delivered satisfactorily, in accordance with certain trust attributes. The system model or architecture with three main components: cloud service provider (CSP); user/rater and Distributed feedback storage (FBS). Under the Intercloud system model, the CSPs provide cloud services collaboratively to users. It automatically collect cloud feedback information from different cloud services and depict a uniform information model. Generate specific configuration solutions by aggregating different cloud resources for target applications based on users' preferences and constraints. Evaluate and recommend cloud solutions.

5.1.1.2 USER ACCESS

The data owner can be the cloud consumer and the cloud provider that owns the cloud within which the cloud service resides. User can access cloud and utilize the resources offered by cloud. Trust offers a rating /reputation based feedback mechanism that being a significant reference for various users. For various cloud environments, dependability of a feedback-mechanism stands highly significant. There may exist numerous malicious users in an open cloud environment and their ratings can result in misleading and false output. With the Feedback provider registered user can be detected by utilizing their login approach and permitting or granting access to only authenticated users.

5.1.1.3 REVIEW ENCRYPTION

After using a cloud service provided by a CSP, a (business/consumer/agent) user registers with the FBS as a rater so that feedback/ratings on the service can be submitted. Admin collect reviews and have various types of reviews. Reviews may be rating reviews, text reviews and smileys reviews. All reviews are stored in database for future evaluation. Ratings, reviews and emoticons are stored in database. Rating, Reviews and Emoticons are the evaluation or assessment of something, in terms of quality (as with a critic rating a novel), quantity or some combination of both. Collected reviews are encrypted using Advanced Encryption Standard (AES) algorithm.

5.1.1.4 REVIEW ANALYSIS

Review analysis refers to the use of natural language processing, text analysis, computational linguistics to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis is widely applied to voice of the customer materials such as reviews and ratings for applications that range from marketing to customer service to buy the products efficiently. Admin can analyze whether the product is positive or negative. In star rating, can calculate star count values. In text reviews, extract keywords and matched with database. Then smileys reviews are calculated based positive and negative symbols.

5.1.1.5 RECOMMENDATION PROCESS

Recommender systems are a subclass of information filtering system that seek to predict the "rating" or

"preference" that a user would give to an item. User can search the cloud server for accessing storage. And view the service provider based on review details. Implement the SVM algorithm to classify the review such as positive or negative. Positive cloud server is display in recommendation panel based on ratings and reviews. If the service provider has negative review means, automatically the positive service provider recommended to the user.

VI. SYSTEM IMPLEMENTATION

One of the most important factors for the adoption and success of cloud computing is trust. Trust evaluation is a fundamental component for secure computing in the cloud environment. In this paper a new trust evaluation method has been proposed. In this scheme, each new customer of a cloud service can determine its trust level using the past experiences of previous customers of this cloud service. The proposed method introduces feedback analysis method that helps distinguish between malicious and reliable customer feedbacks. The proposed scheme assigns a reliability weight to each customer feedback. Users who have shared any cloud service before and did rate it similarly are more likely to have the same opinion of a new service. Hence, their evaluation of a new service should be given a higher weight. In cases where no customer has shared a common service with us before, then existing customers who have had a feedback closer to the majority (in any service) may seem to be more reliable and hence their opinion is given a higher weight. The balance between the feedback of customers with shared service and those of majority consensus in the final decision is determined dynamically and based on the rate of malicious attacks. This way the method can respond better to changes in the rate and sources of malicious attacks. The proposed method has been compared with an existing trust evaluation method and proved its superiority in minimizing the effect of malicious feedbacks, and having a faster response time.

SCOPE

Cloud services are cloud computing resources and applications deployed on the Internet or cloud computing platform, and users can access the required cloud services at any time. However, users face the diversity and complexity of quality of service (QoS) when evaluating and selecting cloud services. Therefore, it is important to study and establish an effective and objective trust model to improve user satisfaction and interaction success rate. In order to ensure the accuracy of direct trust, here proposes a trust evaluation approach based user’s feedback. Feedbacks are collected and analysed to recommend the best cloud services to end users.

6.1 FRONT END: PHP

PHP: Hypertext Preprocessor (the name is a recursive acronym) is a widely used, general-purpose scripting language that was originally designed for web development to produce dynamic web pages. For this purpose, PHP code is embedded into the HTML source document and interpreted by a web server with a PHP processor module, which generates the web page document. As a general-purpose programming language, PHP code is processed by an interpreter application in command-line mode performing desired operating system operations and producing program output on its standard output channel. It may also function as a graphical application. PHP is available as a processor for most modern web servers and as standalone interpreter on most operating systems and computing platforms. PHP was originally created by Rasmus Lerdorf in 1995 and has been in continuous development ever since. The main implementation of PHP is now produced by The PHP Group and serves as the de facto standard for PHP as there is no formal specification. PHP is free software released under the PHP License, which is incompatible with the GNU General Public License (GPL) because restrictions exist regarding the use of the term PHP.

Hypertext refers to files linked together using hyperlinks, such as HTML (Hyper Text Markup Language) files. Preprocessing is executing instructions that modify the output. Below is a demonstration of the difference between HTML and PHP files.

Accessing an HTML Page

1. our browser sends a request to that web page's server (computer) for the file (HTML or image) you wish to view.
2. The web server (computer) sends the file requested back to your computer.
3. Your browser displays the file appropriately.
4. If you request a PHP file (ends with ".php"), the server handles it differently.

Accessing a PHP Page

1. Your browser sends a request to that web page's server for the PHP file you wish to view.
2. The web server calls PHP to interpret and perform the operations called for in the PHP script.
3. The web server sends the output of the PHP program back to your computer.
4. Your browser displays the output appropriately.

Benefit of PHP

Because the server does processing, the output of PHP files changes when its input changes. For example, most of the pages on the Horticulture site have only two (2) PHP commands:

1. Include the header file that defines the links on the left, the banner, and the quick links at the top.
2. Include the footer file that displays the mission statement and Horticulture contact information.

Because including the files is performed every time the PHP file is accessed, when the header/footer files change, the new content will be immediately updated. In other words, if you add a new link, every page that includes the header will immediately display the new link.

Security

About 30% of all vulnerabilities listed on the National Vulnerability Database are linked to PHP. These vulnerabilities are caused mostly by not following best practice programming rules: technical security flaws of the language itself or of its core libraries are not frequent (23 in 2008, about 1% of the total). Recognizing that programmers make mistakes, some languages include taint checking to detect automatically the lack of input validation which induces many issues. Such a feature is being developed for PHP, but its inclusion in a release has been rejected several times in the past. There are advanced protection patches such as Suhosin and Hardening-Patch, especially designed for Web hosting environments.

PHPIDS adds security to any PHP application to defend against intrusions. PHPIDS detects attacks based on cross-site scripting (XSS), SQL injection, header injection, directory traversal, remote file execution, remote file inclusion, and denial-of-service (DoS)

Syntax

The PHP interpreter only executes PHP code within its delimiters. Anything outside its delimiters is not processed by PHP (although non-PHP text is still subject to control structures described in PHP code). The most common delimiters are <?php to open and ?> to close PHP sections. <script language="php"> and </script> delimiters are also available, as are the shortened forms <?or<?= (which is used to echo back a string or variable) and ?> as well as ASP-style short forms <% or <%= and %>. While short delimiters are used, they make script files less portable as support for them can be disabled in the PHP configuration, and so they are discouraged. The purpose of all these delimiters is to separate PHP code from non-PHP code, including HTML.

The first form of delimiters, <?php and ?>, in XHTML and other XML documents, creates correctly formed XML 'processing instructions'. This means that the resulting mixture of PHP code and other markup in the server-side file is itself well-formed XML.

Variables are prefixed with a dollar symbol, and a type does not need to be specified in advance. Unlike function and class names, variable names are case sensitive. Both double-quoted ("") and here-doc strings provide the ability to interpolate a variable's value into the string. PHP treats newlines as whitespace in the manner of a free-form language(except when inside string quotes), and statements are terminated by a semicolon. PHP has three types of comment syntax: /* */ marks block and inline comments; // as well as # are used for one-line comments. The echo statement is one of several facilities PHP provides to output text, e.g., to a Web browser.

In terms of keywords and language syntax, PHP is similar to most high level languages that follow the C style syntax. if conditions, for and while loops, and function returns are similar in syntax to languages such as C, C++, Java and Perl.

Data types

PHP stores whole numbers in a platform-dependent range, either a 64-bit or 32-bit signed integer equivalent to the C-language long type. Unsigned integers are converted to signed values in certain situations; this behavior is different from other programming languages. Integer variables can be assigned using decimal (positive and negative), octal, and hexadecimal notations. Floating point numbers are also stored in a platform-specific range. They can be specified using floating point notation, or two forms of scientific notation. PHP has a native Boolean type that is similar to the native Boolean types in Java and C++. Using the Boolean type conversion rules, non-zero values are interpreted as true and zero as false, as in Perl and C++. The null data type represents a variable that has no value. The only value in the null data type is NULL. Variables of the "resource" type represent references to resources from external sources. These are typically created by functions from a particular extension, and can only be processed by functions from the same extension; examples include file, image, and database resources. Arrays can contain elements of any type that PHP can handle, including resources, objects, and even other arrays. Order is preserved in lists of values and in hashes with both keys and

values, and the two can be intermingled. PHP also supports strings, which can be used with single quotes, double quotes, nowdoc or heredoc syntax.

Functions

PHP has hundreds of base functions and thousands more via extensions. These functions are well documented on the PHP site; however, the built-in library has a wide variety of naming conventions and inconsistencies. PHP currently has no functions for thread programming, although it does support multi-process programming on POSIX systems.

6.2 BACK END: MySQL

MySQL is the world's most used open source relational database management system (RDBMS) as of 2008 that run as a server providing multi-user access to a number of databases. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack—LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL. For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, Joomla, Word Press, phpBB, MyBB, Drupal and other software built on the LAMP software stack. MySQL is also used in many high-profile, large-scale World Wide Web products, including Wikipedia, Google(though not for searches), Image book Twitter, Flickr, Nokia.com, and YouTube.

Inter images

MySQL is primarily an RDBMS and ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. The official set of MySQL front-end tools, MySQL Workbench is actively developed by Oracle, and is freely available for use.

Graphical

The official MySQL Workbench is a free integrated environment developed by MySQL AB, that enables users to graphically administer MySQL databases and visually design database structures. MySQL Workbench replaces the previous package of software, MySQL GUI Tools. Similar to other third-party packages, but still considered the authoritative MySQL frontend, MySQL Workbench lets users manage database design & modeling, SQL development (replacing MySQL Query Browser) and Database administration (replacing MySQL Administrator).MySQL Workbench is available in two editions, the regular free and open source Community Edition which may be downloaded from the MySQL website, and the proprietary Standard Edition which extends and improves the feature set of the Community Edition.

Web Clients

A Web customer comprises of two sections:

1. Dynamic Web pages containing different sorts of markup dialect which are created by Web parts running in the Web level.
2. Web program, which renders the pages got from the server.

A Web customer is now and again called a slim customer. Slim customers as a rule don't question databases, execute complex business guidelines, or associate with legacy applications.

Within SQL, we have two forms of languages. These forms differ in that one is used to build and edit the structure of the database while the other is used to create and edit the actual data within the database. These two languages are known as data definition language and data manipulation language.

Data Definition Language (DDL)

Data definition language is one of the subcategories of SQL. It is used to define and work with the database schema (structure). This includes the attributes (columns) within each table, the name of each table, the name of the database, and the connection of keys between tables. Here are general explanations of the types of commands in DDL:
CREATE – used to create the database, the tables, and the columns within each table. Within the create statement we also define the data type of each column. A data type is

literally the type of data we are supposed to store within each column, whether it be an integer, a date, or a string.

ALTER – used to alter existing database structures. This includes adding columns and more.
RENAME – This is used to…rename.
DROP – This is used to destroy your database or table.

Data Manipulation Language (DML)

Data manipulation language is used to work with the actual data within the database. if we looked at an example with a users table, the table is created with DDL while the value "Caleb Curry" is entered using DML.

The main statement in DML are:

SELECT – this is used to select data from our database. We first say SELECT and then we say what columns to select. After we say what columns, we specify what tables using FROM. After we select what columns and what tables we can limit our results using a WHERE clause.
INSERT INTO – This is used to insert new values.
UPDATE – This is used to change values.
DELETE – this is used to delete values (the database structure stays the same, only inserted values are removed).

HTML

HTML is a markup language for describing web documents (web pages).

1. Hyper is the opposite of linear. It used to be that computer programs had to move in a linear fashion. This before this, this before this, and so on. HTML does not hold to that pattern and allows the person viewing the World Wide Web page to go anywhere, any time they want.
2. Text is what you will use. Real, honest to goodness English letters.
3. Mark up is what you will do. You will write in plain English and then mark up what you wrote. More to come on that in the next Primer.
4. Language because they needed something that started with " L " to finish HTML and Hypertext Markup Louie didn't flow correctly. Because it's a language, really but the language is plain English.

HTML remains for Hyper Text Markup Language. It is a basic content designing dialect used to make hypertext records. It is a stage free dialect not at all like most other programming dialect. HTML is impartial and can be utilized on numerous stage or desktop. It is this component of HTML that makes it mainstream as standard on the WWW.

This adaptable dialect permits the making of hypertext connections, otherwise called hyperlinks. These hyperlinks can be utilized to unite reports on diverse machine, on the same system or on an alternate system, or can even indicate purpose of content in the same record.

HTML is utilized for making archives where the accentuation is on the presence of the record. It is likewise utilized for DTP. The records made utilizing HTML can have content with diverse sizes, weights and hues. It can also contain graphics to make the document more effective.

VII. CONCLUSION

In this work proposed a novel trust evaluation model. This distributed protocol provides some distinctive features, particularly for the Intercloud environment. First, it supports user anonymity by means of blind signature, facilitating users to provide honest feedback without fear of a retaliatory attack. Second, by means of an innovative mechanism for storing feedback, feedback privacy can be protected by using homomorphic encryption with verifiable secret sharing. Third, it allows customized processing of evaluation results while protecting feedback privacy.

VIII. FUTURE ENHANCEMENT

In future, we are going to use an approach to filter feedbacks from malicious customers. Also, model will be extended to work in scenario where multiple cloud service providers are inter-connected.

REFERENCES

[1] Liu, Li, Shuxian Gu, Dongmei Fu, Miao Zhang, and Rajkumar Buyya. "A New Multi-objective Evolutionary Algorithm for Inter-Cloud Service Composition." TIIS 12, no. 1 (2018): 1-20.
[2] Sotiriadis, Stelios, Nik Bessis, Euripides GM Petrakis, Cristiana Amza, Catalin Negru, and Mariana Mocanu. "Virtual machine cluster mobility in inter-cloud platforms." Future Generation Computer Systems 74 (2017): 179-189.
[3] Barlaskar, Esha, Peter Kilpatrick, Ivor TA Spence, and Dimitrios S. Nikolopoulos. "MyMinder: A User-centric Decision Making Framework for Intercloud Migration." In CLOSER, pp. 560-567. 2017.
[4] Noor, Talal H., Quan Z. Sheng, Lina Yao, Schahram Dustdar, and Anne HH Ngu. "CloudArmor: Supporting

reputation-based trust management for cloud services." IEEE transactions on parallel and distributed systems 27, no. 2 (2015): 367-380.

[5] Clark, Michael R., Kyle Stewart, and Kenneth M. Hopkinson. "Dynamic, privacy-preserving decentralized reputation systems." IEEE Transactions on Mobile Computing 16, no. 9 (2016): 2506-2517.

[6] S. Sotiriadis and N. Bessis, "An inter-cloud bridge system for heterogeneous cloud platforms," Future Generation Comp. Syst., vol. 54, pp. 180–194, 2016.

[7] A. Schaub, R. Bazin, O. Hasan, and L. Brunie, "A trustless privacy preserving reputation system," IACR Cryptology ePrint Archive, vol. 2016, p. 16, 2016.

[8] B. Chang, Y. Lee, and Y. Liang, "Reward-based markov chain analysis adaptive global resource management for inter-cloud computing," Future Generation Comp. Syst., vol. 79, pp. 588–603, 2018.

[9] C. Ngo, Y. Demchenko, and C. de Laat, "Toward a dynamic trust establishment approach for multi-provider intercloud environment." in Proc. 4th Int. Conf. Cloud Comput. Technol. and Sci., 2012, pp. 532–538.

[10] A. Schaub, R. Bazin, O. Hasan, and L. Brunie, "A trustless privacy preserving reputation system," IACR Cryptology ePrint Archive, vol. 2016, p. 16, 2016.