

SIMULADOR PARA MINIMIZAR LA VIOLACIÓN DE ACUERDOS DE NIVELES DE SERVICIO (SLAs) EN ARQUITECTURAS DE NUBES BASADAS EN MECANISMOS DE MERCADO

Erica Castro*, José Sumba**, Carlos Mera***

Facultad de Ingeniería Eléctrica y Computación(FIEC)
Escuela Superior Politécnica del Litoral(ESPOL)
Campus Gustavo Galindo, Km 30.5 vía Perimetral
Apartado 09-01-5863. Guayaquil-Ecuador
eriescas@espol.edu.ec, jlsumba@espol.edu.ec, cjmera@espol.edu.ec

Resumen

El presente proyecto consiste en diseñar un modelo que permita minimizar las violaciones a un SLA (Service Level Agreement por sus siglas inglés) dentro de una arquitectura en Nube basada en un mecanismo de mercado. Se denomina violación cuando el proveedor no cumple con los requerimientos acordados con el usuario dentro del SLA.

Sin duda alguna, minimizar las violaciones con la finalidad de garantizar el cumplimiento de los parámetros QoS ocupa un lugar clave dentro de la relación cliente-proveedor. En consecuencia, se realizará un análisis de mecanismos de mercado y sistemas de reputación existentes con el objetivo de diseñar un modelo que permita disminuir tales violaciones. Dentro de la etapa de diseño del modelo, el uso del mecanismo de mercado y el sistema de reputación ofrecerá al usuario la oportunidad de seleccionar a un proveedor, no sólo considerando el precio del servicio, sino también la reputación del proveedor, con lo cual se espera que los servicios sean alojados por proveedores con una reputación alta y cuya probabilidad de falla sea baja.

La etapa de experimentación del modelo se realizará en un simulador implementado sobre el framework de CloudSim. Los datos para la simulación al igual que los resultados serán manejados en formato XML. Además, se presentarán los resultados gráficamente con la librería JFree-Chart para facilitar la interpretación de los mismos.

Finalmente, el uso del lenguaje estadístico R permitirá procesar los resultados y generar gráficas de las violaciones ocurridas en la simulación para los mecanismos de mercado y sistemas de reputación previamente seleccionados.

Abstract

The present project consists of the design of a model to analyze violations on SLAs (Service Level Agreements) within a Cloud architecture, based on market mechanisms. A violation occurs when the provider does not meet the requirements established in the SLA.

Minimizing violations in order to guarantee fulfillment of Service Level Agreements is indeed key to ensure a positive client-provider relationship. Therefore, market mechanisms and reputation systems analyses will be conducted in order to design a model to reduce the number of such violations. During the design model stage, the use of a market mechanism and the reputation system will allow the user to choose a provider, not only considering the price of the service, but its reputation. It is therefore expected that services be allocated with providers that have a higher reputation and a lower fail rate.

The experimentation stage will be conducted in a simulator implemented over the CloudSim framework. Data for the simulation and simulation results will be handled using XML format. Furthermore, results will be presented through graphs created with the JFree-Chart library to facilitate its interpretation.

Finally, R statistical language will be used to process the results obtained and generate charts of the violations occurred during the simulation for the market mechanisms and reputation systems previously selected.

1. INTRODUCCIÓN

EN la Nube existen distintos modelos de servicios: software como servicio (SaaS), plataforma como servicio (PaaS) e infraestructura

como servicio (IaaS)[7]. Los proveedores cobran por dichos servicios de acuerdo al uso y ofrecen la posibilidad de escalar dinámicamente según la demanda. Ellos deben garantizar que los servicios estén disponibles y cumplan con los requerimientos

* eriescas@espol.edu.ec

** jlsumba@espol.edu.ec

*** cjmera@espol.edu.ec

del usuario establecidos en el contrato denominado SLA [4]. Para esto, requerimientos funcionales y no funcionales son definidos en dicho contrato. Adicionalmente, en él se establecen las obligaciones tanto del usuario como del proveedor y las sanciones en caso de incumplimiento. El incumplimiento del contrato es conocido como una violación al SLA.

Se analizarán algunos de los mecanismos de mercado existentes, los cuales se basan en el precio para la selección de proveedores y la asignación de recursos. De los mecanismos a ser analizados, se seleccionarán dos: subasta de oferta publicada (Posted Offer) y subasta inversa (Reverse Auction). Implementando este mecanismo en la Nube permitiría que el proveedor con la mejor oferta y el precio más bajo para el usuario sea el que aloje el servicio, pero esto no garantizaría que se eviten violaciones al SLA. Por lo tanto, con el propósito de minimizar tales violaciones se hará uso de indicadores de desempeño y confiabilidad del proveedor, los cuales se denominan reputación. La reputación puede ser definida por métricas que se basan en el historial y en la calificación que los usuarios dieron a los servicios alojados. Del mismo modo, serán analizados algunos sistemas de reputación existentes y se escogerán dos para ser utilizados en nuestro proyecto: Blurred Beta Reputation y Beta Based Reputation. Adicionalmente, propondremos una nueva métrica: Average Combined Reputation. Ésta reputación la hemos denominado Metric Based Reputation porque esta basada en métricas. Por otro lado, la reputación también puede ser definida por el cumplimiento de los parámetros QoS (Quality of Service) establecidos en el SLA. Dichos parámetros son indicadores de calidad y para el presente proyecto se utilizarán cuatro: Availability, Reliability, Performance y Cost. Con la finalidad de obtener una única reputación resultado de la combinación de dichos QoS se empleará un método conocido como Logic Scoring of Preferences (LSP) descrito en [9], el cual permitirá unificar dichos criterios y hallar un valor único. A ésta reputación la hemos denominado QoS Based Reputation.

En nuestros días, realizar estos experimentos en plataformas de Nubes reales requiere tiempos extensos y resulta costoso. Además, la configuración de escenarios y la replicación de los mismos para la experimentación resulta costoso en términos de tiempo. Por tales motivos hemos optado por implementar un simulador sobre el framework de CloudSim que cuenta con componentes para modelar y simular la Nube de una manera relativamente sencilla y de bajo costo.

Durante el proceso de simulación los datos de usuarios, proveedores, servicios a alojarse en la Nube y resultados serán manejados utilizando archivos XML. En el simulador se permitirá importar y exportar los datos en dicho formato. Además, los resultados obtenidos serán representados gráficamente a fin de facilitar su interpretación. Posteriormente, dichos resultados serán procesados con el lenguaje R para generar las tablas y curvas de las violaciones ocurridas en simulación con el fin de obtener evidencia de la disminución de dichas violaciones a lo largo de los diferentes experimentos.

2. ANTECEDENTES Y TRABAJO RELACIONADO

Los estudios e investigaciones acerca de Cloud Computing son diversos, pero para nuestra investigación nos enfocaremos en los elementos principales de los entornos Cloud y sus conceptos. Además aclararemos términos imprescindibles en nuestro estudio como: sistema basado en mercado, Nubes basadas en mercado, reputación, entre otros. Los cuales se describen a continuación:

- **Mecanismos de Mercado:** la Economía Experimental consiste en la evaluación sistemática de las teorías económicas (destinada a explicar la actividad del mercado) con el fin de probar o refutar dichas teorías mediante la experimentación en ambientes controlados dio lugar a mecanismos, como la subasta, que son utilizados en la Nube con la finalidad de encontrar nuevos métodos para la selección de proveedores para el alojamiento y ejecución de servicios Smith [6].
- **Sistema de Reputación:** los esquemas de calificación (rating) proporcionan información acerca de la calidad percibida y la utilidad de los contenidos en línea, mientras que por su parte los sistemas de reputación brindan a los usuarios información sobre el comportamiento de los proveedores de servicios en transacciones pasadas. En el caso de software que usen sistemas de reputación existen: Mojo y Karma, éstos pueden mejorar el ambiente de interacción en línea, ayudando a los usuarios a decidir en quién confiar y promover una conducta digna de confianza [2]. Existen diversos sistemas de reputación como: acumulativos, promedio, blurred, onlyLast, eigenTrust, adaptativos y beta.
- **SLA en arquitecturas de Nube:** es un contrato entre usuario y proveedor donde se re-

gistra el nivel de servicio especificado por varios atributos, tales como: disponibilidad, capacidad de servicio, rendimiento, funcionamiento, facturación e incluso en éste se incluyen las sanciones en caso de que se incurra en una violación del mismo [4].

- **QoS:** la calidad de servicio es un término extenso, ampliamente utilizado y que esta relacionado a la experiencia general que tiene un usuario al estar en contacto con un servicio Cloud o en base al nivel de calidad que muestra una aplicación que usa una arquitectura de Nube [5].
- **CloudSim:** es un framework de simulación generalizado y extensible con un enfoque de simulación de eventos [3]. Presenta las siguientes características:
 - Soporte para el modelamiento y simulación de la Nube.
 - Una plataforma autónoma para el modelamiento servicios, brokers y políticas de asignación.
 - Probar y evaluar las aplicaciones en un entorno repetible y controlable.
 - Ajustar el sistema a escenarios críticos.
 - Experimentar con diferentes cargas de trabajo para evaluar las políticas de asignación.

La computación en la Nube es un ejemplo representativo de un sistema basado en mercado por la competencia existente entre proveedores para alojar servicios solicitados por usuarios. Investigaciones realizadas por [1], [11] y [12] se enfocan en combinar principios de economía para encontrar soluciones a problemas en la Nube buscando: mejorar la asignación de recursos, establecer mecanismos de control y prevenir la violación de SLA. En la sección posterior se describirá de forma detallada la arquitectura del simulador y los componentes utilizados.

3. ARQUITECTURA DEL SIMULADOR

La figura 1 muestra los componentes utilizados para la asignación de recursos de proveedores de servicios a diferentes usuarios de una Nube basada en un mecanismo de mercado. Dicha asignación se realizó buscando cumplir un SLA previamente definido. Con la finalidad de garantizar una asignación adecuada, este sistema utiliza un mecanismo de mercado que considera el costo del servicio y

la reputación del proveedor como factores importantes al momento de escoger el proveedor al que debé ser asignado el servicio. Dado que nuestro objetivo es reducir las violaciones de SLAs por parte de los proveedores se decidió que la reputación del proveedor dependería de los servicios que éste pudiera alojar de manera exitosa.

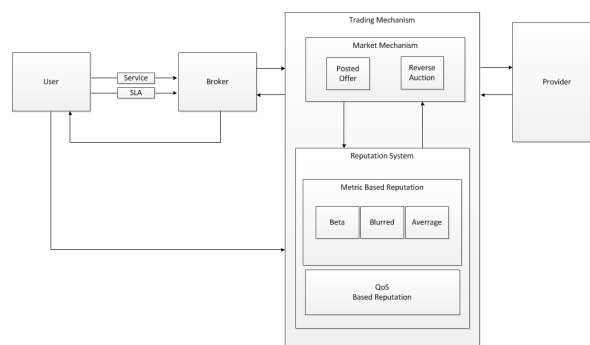


Figura 1: Componentes del modelo

3.1. Sistema de reputación

Siendo éste uno de los principales componentes de nuestro sistema, optamos por realizar el cálculo de la reputación usando dos tipos de algoritmos: Metric Based Reputation y QoS Based Reputation. Buscando de esta manera recabar evidencia que pruebe que el número de violaciones por proveedor se reduce si se toma en cuenta la reputación en el proceso de selección de éste independientemente del tipo de algoritmo utilizado para establecer su reputación. Además, nos permitirá analizar si existe una diferencia significativa en el número de violaciones por algoritmo. A continuación, se detallan las características y diferencias de los algoritmos implementados:

I. **Metric Based Reputation:** denominamos Metric Based Reputation a la reputación que es calculada con un algoritmo que considera el rating asignado por los usuarios a los servicios alojados en el proveedor. En nuestro caso particular fueron seleccionadas dos métricas existentes: Beta y Blurred, pero además como aporte decidimos combinar estas métricas usando una fórmula de promedio, obteniendo así una nueva métrica a la que llamamos **Average**.

a.) **Beta Based Reputation:** se basa en una métrica que predice el comportamiento de un agente, en nuestro caso del proveedor en su próxima transacción en base a su comportamiento anterior. Catalogando

las acciones de dicho proveedor como acciones buenas r y acciones malas s , las cuales se combinan mediante el uso de las siguientes fórmulas, las cuales fueron tomadas de[10]:

Primero, se debe establecer $r(a)$ que representa el valor de las acciones buenas realizadas por el proveedor:

$$r^a = \sum_{i=1}^{\#E_a} \lambda^{\#(E_a) - \frac{i(1 + \rho(a, E_a[i]))}{2}} \quad (1)$$

y las malas acciones $s(a)$ así;

$$s^a = \sum_{i=1}^{\#E_a} \lambda^{\#(E_a) - \frac{i(1 - \rho(a, E_a[i]))}{2}} \quad (2)$$

Con lo cual la reputación del proveedor basada en esta métrica denotado por $a \in A$ queda definida por la siguiente ecuación tomada de[10]:

$$r(a)_{Beta} = \frac{r^a - s^a}{r^a + s^a + 2} \quad (3)$$

donde $0 \leq \lambda \leq 1$

- b.) **Blurred Beta Reputation:** el uso de esta reputación permite cálculos que no dependen del tiempo, pero donde se consideran todos los trabajos realizados por el proveedor pero se les da una mayor ponderación al rating de los trabajos más recientes. Asumiendo que es **altamente probable** que dicho proveedor se comporte como lo hizo en sus transacciones más recientes. Basado en este supuesto. La reputación de un proveedor $a \in A$ viene dado por:

$$r(a)_{Blurred} = \sum_{i=1}^{\#E_a} \frac{\rho(a, E_a[i])}{\#(E_a) - i + 1} \quad (4)$$

donde los valores que el rating ρ puede tomar están definidos por el siguiente conjunto de valores: $\{-1, 0, 1\}$, lo cual fue definido en[10].

- c.) **Average Combined Reputation:** esta métrica ha sido definida por nosotros mediante la combinación de las dos métricas anteriores, realizando un promedio entre las dos con la finalidad de mejorar la reputación tomando en cuenta tanto las transacciones más recientes del proveedor como a su vez todo su historial y así poder

asignarle una reputación acorde a su comportamiento. Por lo tanto la reputación del proveedor ha quedado definida por $a \in A$:

$$r(a)_{Average} = \frac{r(a)_{Beta} + r(a)_{Blurred}}{2}, \quad (5)$$

- II. **QoS Based Reputation:** para establecer el cálculo de esta reputación, se consideró necesario definir los parámetros QoS, los cuales fueron acordados en el SLA y que para este proyecto son: *Availability, Reliability, Performance* y *Cost*.

Una vez establecidos los parámetros a ser medidos y dado que dichos QoS están en unidades distintas, buscamos una forma de obtener una única reputación resultado de la combinación de dichos QoS en base a un método conocido como Logic Scoring of Preferences (**LSP**) descrito en[9]. Dicho método nos permite construir una sola reputación en base a otros parámetros sin importar las unidades en las que están medidos dichos parámetros. Éste método funciona de la siguiente manera: Primero se debe calcular el L definido mediante la siguiente ecuación tomada de[9]:

$$L = (|\omega_1|E_r^1 + |\omega_2|E_r^2 + \dots + |\omega_n|E_r^n)^{1/r};$$

$$0 \leq E \leq 1, \sum_{n=1}^n |\omega_i| = 1 \quad (6)$$

donde E está definida en base a tres métricas distintas, las cuales consideran el tipo de dato del parámetro QoS a evaluar:

- a. Tipo de dato numérico: si el parámetro QoS puede ser expresado con un tipo de dato numérico. Entonces la función de evaluación E definida en[9] es:

$$E = \begin{cases} 1 - \left(\frac{v_{max} - v}{v_{max} - v_{min}} \right) & ; \text{si } \omega \geq 0 \\ & ; \text{si } \omega \leq 0 \\ \left(\frac{v_{max} - v}{v_{max} - v_{min}} \right) & ; \text{para el resto} \end{cases} \quad (7)$$

Donde v_{max} representa el valor máximo de todos los servicios y v_{min} el valor mínimo. v es el valor del servicio a evaluar. Nótese que se considera el peso (ω) del parámetro QoS ha evaluar. El significado del peso (ω) es el siguiente: si el peso es igual a 1, entonces el criterio es requisito duro (**HARD**), lo

que significa que los servicios que no cumplan con este criterio deben ser desechados. Si el peso(ω) es menor que 1 y mayor que 0, entonces el criterio se considera un requisito suave(SOFT) y se ejecuta el servicio.

- b. Tipo de dato booleano: Si el parámetro QoS es de tipo de dato booleano. Entonces la función de evaluación E definida en[9] es:

$$E = \begin{cases} 1 & \text{;si el parámetro QoS es met} \\ 0 & \text{;para el resto} \end{cases} \quad (8)$$

- c. Otros tipos de datos: Si el parámetro QoS es de tipo de dato definido por el usuario. Entonces la función de evaluación E definida en[9] es:

$$E = \frac{(e_1 + e_1 + \dots + e_n)}{n} \quad (9)$$

donde e es el valor
para cada elemento

donde ω representa el peso de cada parámetro, y r es el valor lógico adoptado del método LSP, se optó por usar $r=1$. A continuación, presentamos las definiciones de los parámetros QoS tomadas de[8], en la que se basa el presente proyecto:

1. **Availiability:** Este requisito está relacionado con la capacidad del servicio de responder las peticiones del usuario cuando éste lo requiera.

$$a = \frac{\text{TotaldelTiempodelServicio} - \text{TiempoDeServicioCaido}}{\text{TotaldelTiempodelServicio}}$$

2. **Reliability:** Este requisito está relacionado con la cantidad de servicios realizados con éxitos y el total de servicios ejecutados.

$$r = \frac{\#serviciosExitosos}{\text{totaldeServicios}}$$

3. **Performance:** Este requisito está relacionado con el monto de los recursos y el tiempo de ejecución de los servicios. En la Nube tiene que poseer un alto rendimiento por lo que a menor tiempo del servicio menor cantidad de recursos asignados en la ejecución de una petición.

4. **Cost:** Este requisito está relacionado con la cantidad que el usuario está dispuesto a pagar con el fin de estar provisto de un servicio que es compatible con todos los requisitos no funcionales adicionales.

4. DISEÑO E IMPLEMENTACIÓN DEL SIMULADOR

Los principales componentes que forman parte del diseño del presente simulador son mostrados en la figura 2.

- **XMLLoader:** este componente usa DOM(Document Object Model) por sus siglas en inglés. Nos permite manejar una estructura para los archivos utilizados en la herramienta de simulación. Los escenarios que contienen la información de proveedores, usuarios y servicios a solicitar son guardados por medio de este componente en archivos XML. De igual manera los resultados de la simulación, los servicios por proveedor que fueron alojados exitosamente o no, también son guardados utilizando este componente. El formato utilizado, XML, facilita la lectura del contenido y el posterior análisis de los resultados en R para éstos archivos no resultó complicado.
- **CloudSim:** según lo revisado anteriormente es el componente que nos permite realizar la simulación.
- **ResultsGenerator:** componente encargado de manejar la generación de los resultados de la simulación, los mismos que se muestran en logs que registran los eventos ocurridos en la simulación. Pero, para generar reportes que permitan una mejor visualización de los mismos y un análisis más rápido se utilizó JFreeChart durante la implementación de este componente.

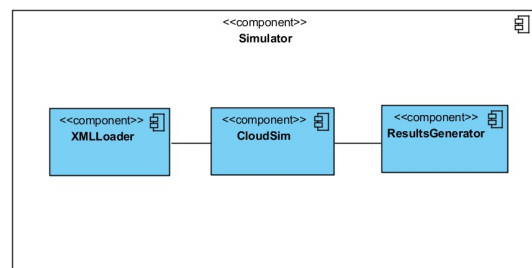


Figura 2: Diagrama de Componentes

5. EXPERIMENTOS

Los experimentos realizados tienen como objetivo principal mitigar las violaciones en los parámetros QoS de los SLAs, con lo cual esperaríamos

obtener como resultado de la simulación una disminución en el número de violaciones por proveedor independientemente del tipo de mecanismo de mercado utilizado durante la misma. Para cada simulación se definió lo siguiente:

- I. Mecanismo de mercado: puede ser Posted Offered o Reverse Auction.
- II. Sistema de reputación: puede usar métricas como Beta System, Blurred System, Average System o puede basarse en los parámetros QoS(QoS Based System).

El proceso de simulación para el mecanismo de mercado y el sistema de reputación consistió en:

- I. Definir 4 proveedores.
- II. Definir 10 servicios a alojar en los 4 proveedores.
- III. Simular el alojamiento de servicios. Cada simulación constituye lo que se denominó ronda, para la siguiente ronda se actualiza el estado de los proveedores, la reputación se verá afectada. Un total de 10 rondas fueron ejecutadas.

Dicho proceso se denominó experimento. Se ejecutaron un total de 10 experimentos. Durante la etapa final de nuestra experimentación se realizó una suma de violaciones por ronda. Finalmente, se utilizó el programa estadístico R para realizar el procesamiento de los resultados, obtener el número de violaciones ocurridas por simulación y generar gráficos que permitan analizar e interpretar los resultados obtenidos .

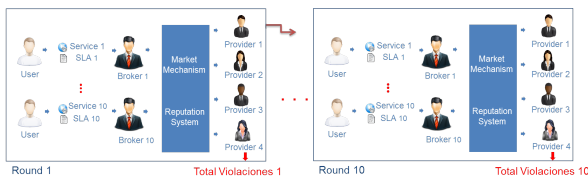


Figura 3: Proceso de experimentación

6. RESULTADOS

De los experimentos para los mecanismos de mercado y sistemas de reputación descritos en la sección previa se obtuvieron los siguientes resultados.

Cuadro 1: Resultados Posted Offered

Posted Offered	Beta	Blurred	Average
Ronda 1	29	23	33
Ronda 2	10	15	4
Ronda 3	9	14	1
Ronda 4	3	12	33
Ronda 5	8	10	11
Ronda 6	4	3	9
Ronda 7	8	3	4
Ronda 8	4	1	10
Ronda 9	9	2	6
Ronda 10	6	7	8

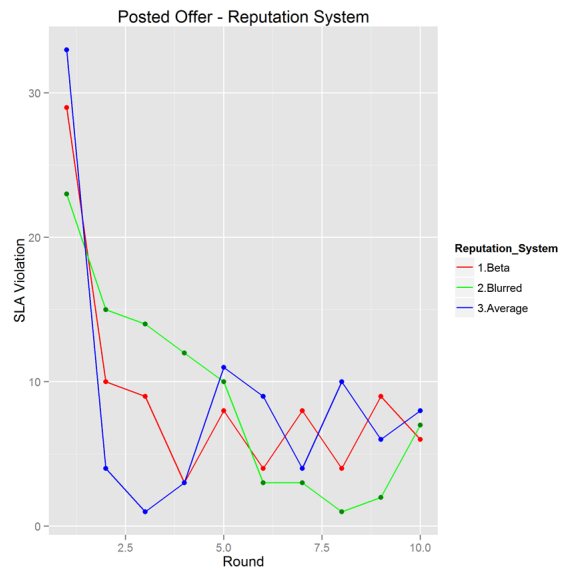


Figura 4: Posted Offered

En el cuadro 1 podemos observar que el número de violaciones tiene una tendencia a disminuir conforme aumenta el número de rondas. En el caso de Posted Offer usando Beta System se tiene que en la *ronda*₁ hay un total de 29 violaciones mientras que en la *ronda*₁₀ hay 6, con lo cual podemos comprobar que hay una disminución considerable en el número de violaciones ocurridas. En cuanto a Blurred System usando el mismo mecanismo de mercado ocurre casi lo mismo: en la primera *ronda*₁ hay un total de 23 violaciones mientras que en la *ronda*₁₀ hay 7. Mientras que con Average System en *ronda*₁ hay un total de 33 violaciones y en la *ronda*₁₀ hay un total de 8. Se puede notar que, usando Average System en la *ronda*₁ se obtiene el mayor número de violaciones y que con Blurred System en la misma ronda se obtiene el menor número de violaciones de los tres sistemas de reputación

usados. Por otra parte, en la *ronda*₁₀ haciendo uso de Beta System se obtiene la mayor disminución en el número de violaciones y con Average System la menor disminución. En la figura 4 queda ilustrado lo descrito anteriormente.

Cuadro 2: Resultados Reverse Auction

Reverse Auction	Beta	Blurred	Average
Ronda 1	27	29	38
Ronda 2	8	8	6
Ronda 3	8	11	5
Ronda 4	7	9	5
Ronda 5	5	5	8
Ronda 6	3	6	4
Ronda 7	6	7	7
Ronda 8	5	3	6
Ronda 9	2	8	4
Ronda 10	1	6	6

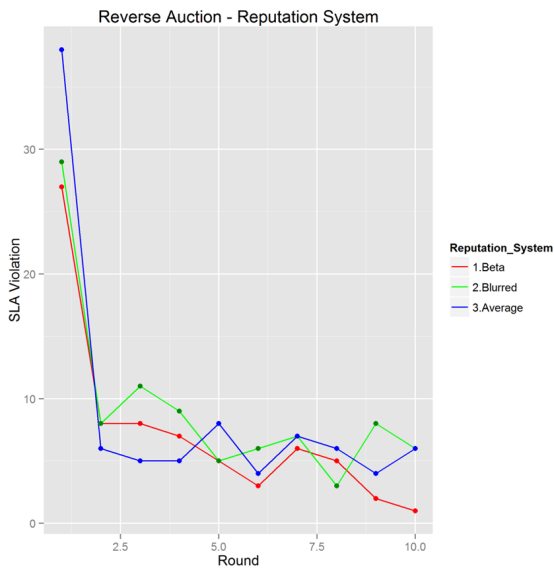


Figura 5: Reverse Auction

Para el mecanismo de mercado Reverse Auction los resultados obtenidos se muestran en el cuadro 2 y se ilustran en la figura 5. Podemos observar una disminución de las violaciones en los SLAs, obteniendo con Beta System en la *ronda*₁ un total de 27 violaciones y en la *ronda*₁₀ un total de 1 violación. En tanto que para Blurred System para la *ronda*₁ existieron un total de 29 violaciones y para la *ronda*₁₀ ocurrieron 6 violaciones en total. Finalmente, para Average System en la *ronda*₁ ocurrieron un total de 38 violaciones y en la *ronda*₁₀ existieron apenas en total 6. Las curvas de la figura 5 muestran los resultados usando Reverse Auction

con las tres métricas establecidas. Sin duda alguna, dicha figura refleja una disminución considerable en el número de violaciones ocurridas, demostrando que durante la *ronda*₁ con Average System se obtiene el mayor número de violaciones mientras que con Beta System el menor. Del mismo modo, en la *ronda*₁₀ el menor número de violaciones ocurre usando Beta System y el mayor número usando las otras dos métricas

Cuadro 3: Resultados QoS Based Reputation System

QoS Based	Posted Offered	Reverse Auction
Ronda 1	29	25
Ronda 2	4	9
Ronda 3	4	10
Ronda 4	7	7
Ronda 5	5	6
Ronda 6	8	4
Ronda 7	7	8
Ronda 8	7	9
Ronda 9	10	7
Ronda 10	4	3

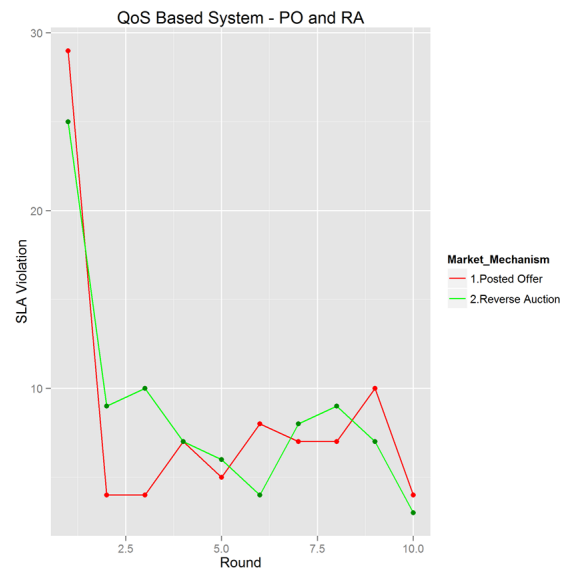


Figura 6: QoS Based Reputation System

Se observa que el mayor número de violaciones en la *ronda*₁ ocurre utilizando Posted Offer; en total existieron 29 violaciones en este escenario. Mientras que en la *ronda*₁₀ el menor número de violaciones que en este caso fue 3 ocurrió usando Reverse Auction. En la curva 6 se ilustra lo que se mencionó en el párrafo anterior, demostrando una disminución considerable en el número de violaciones.

En esta sección se puede observar que en las curvas inicialmente se tiene un gran número de violaciones, luego de ejecutar la primera asignación de servicios y haber modificado la reputación de los proveedores, el número de violaciones disminuye. En las siguientes simulaciones el número de violaciones aumentan y disminuyen, pero a nivel general podemos decir que tienden a disminuir.

En las curvas que corresponden al mecanismo de mercado Reverse Auction para los sistemas de reputación: Beta System, Blurred System, Average System y QoS Based Reputation observamos que la pendiente de las curvas entre ronda y ronda en su mayoría es negativa, con lo cual la disminución de violaciones es más notoria. Cabe recalcar que, en este mecanismo el proceso de selección se realiza con varias ofertas, lo cual hace que el nivel de competencia entre proveedores ocurrido en este mecanismo sea mayor al nivel que existe en Posted Offer.

7. CONCLUSIONES Y TRABAJO A FUTURO

Durante el desarrollo del presente proyecto hemos buscado experimentar con un nuevo modelo para disminuir el número de violaciones a los SLAs que combine los conceptos provenientes de los sistemas basados en el mercado y los sistemas de reputación con la finalidad de mejorar la asignación de recursos en la Nube. Para alcanzar este objetivo hemos implementado el modelo y creado un ambiente controlado a través de una herramienta de simulación de uso abierto (CloudSim). El análisis de los resultados obtenidos a partir de las simulaciones realizadas en esta investigación demuestran que las violaciones a los SLAs disminuyen considerablemente de acuerdo al número de rondas ejecutadas independientemente del sistema de reputación y del mecanismo de mercado empleados para la simulación.

Adicionalmente, hemos podido contrastar dos mecanismos comerciales opuestos uno con el poder de mercado centrado en el vendedor como Posted Offer y otro en el comprador como Reverse Auction, en ambos casos hemos podido contraponer el uso de métricas de reputación con dos enfoques distintos: uno basado en el historial de transacciones del proveedor y el otro basado en los QoS acordados en el SLA. Por otra parte, el crecimiento de los entornos Nube ha incrementado el uso de las aplicaciones de dicho entorno, cabe recalcar que hemos podido comprobar que la reputación de un proveedor es un factor importante a considerar para escogerlo como candidato para usar su servicio, dicho valor permite segmentar o diferenciar a

los buenos proveedores de los malos. Sin embargo, el número de violaciones a los SLAs se ha reducido considerablemente independientemente de lo antes mencionado.

A pesar de que nuestro estudio considera mecanismos de mercado con el poder de mercado centrado en el comprador y en el vendedor, una investigación futura podría implementar otro mecanismo de mercado como doble subasta, que trata de equilibrar el poder. Otro de los aspectos que podrían enriquecer el diseño es la inclusión de nuevos criterios que involucren tiempos, por ejemplo: fecha límite para la finalización del trabajo. En cuanto a la fórmula para establecer la relación entre los distintos parámetros de calidad, esto ha sido una primera propuesta utilizada para fines de experimentación. Podría mejorarse teniendo en cuenta nuevos factores o utilizando diferentes fórmulas en función de otros parámetros clave. Desde el punto de vista de la experimentación, se puede continuar trabajando para ampliar los conceptos de los agentes usuario y proveedor, para que la complejidad de los escenarios aumente, logrando así obtener situaciones más similares a la realidad del mercado, en el cual cuando un proveedor no tiene la demanda que espera o su desempeño no es totalmente eficiente realiza mejoras en los servicios a fin de captar nuevos usuarios, haciendo que los proveedores mejoren sus servicios dinámicamente. Por otro lado, con respecto al origen de los datos para la experimentación a futuro se recomienda el uso de datos estadísticos de entornos reales de este tipo. Además, se podrían implementar mecanismos de comisiones para el agente broker, para posteriormente hacer estudios estadísticamente exhaustivos que determinen bajo qué mecanismo de mercado y sistema de reputación considerando nuevos parámetros y políticas ocurren mejores asignaciones de recursos. Por último, la herramienta podría extenderse a ser utilizada para la composición de servicios para Nubes Federadas.

REFERENCIAS

- [1] ARDAGNA, D., PANICUCCI, B. y PASSACANTANDO, M. A game theoretic formulation of the service provisioning problem in cloud systems. En *Proceedings of the 20th International Conference on World Wide Web, WWW '11*, páginas 177–186. ACM, New York, NY, USA, 2011. ISBN 978-1-4503-0632-4.
- [2] AUTORES, V. Trust building online: Rating and reputation systems. consultado en: <https://depts.washington.edu/>

- ccce/digitalMedia/rebs.html, 2002. Accedido el:02-01-2015.
- [3] CALHEIROS, R. N., RANJAN, R., BELOGLAZOV, A., DE ROSE, C. A. F. y BUYYA, R. Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience*, vol. 41(1), páginas 23–50, 2011. ISSN 1097-024X.
- [4] EMEAKAROHA, V. C., CALHEIROS, R. N., NETTO, M. A., BRANDIC, I. y DE ROSE, C. A. Desvi: An architecture for detecting sla violations in cloud computing infrastructures. En *Proceedings of the 2nd International ICST Conference on Cloud Computing (CloudComp10)*. 2010.
- [5] GANGHISHETTI, P. y WANKA, R. Quality of service design in clouds. 2011.
- [6] KETCHAM, J., SMITH, V. L. y WILLIAMNS, A. W. A comparison of posted-offer and double-auction pricing institutions. *The Review of Economic Studies*, vol. 51(4), páginas 595–614, 1984.
- [7] MELL, P. y GRANCE, T. The NIST definition of cloud computing. *Technical report, Information Technology Laboratory*, vol. 2009(3), 2009.
- [8] ODEH, Y. y ODEH, M. A new classification of non-functional requirements for service-oriented software engineering. 2011.
- [9] REIFF-MARGANIEC, S., YU, H. Q. y TILLY, M. Service selection based on non-functional properties. En *Service-Oriented Computing-ICSOC 2007 Workshops*, páginas 128–138. Springer, 2009.
- [10] SHEN, Y. *Research report: Reputation system simulation results*. Proyecto Fin de Carrera, University of Helsinki, Helsinki Institute for Information Technology (HIIT), Helsinki–Finland, 2009.
- [11] SHI, W. y HONG, B. Resource allocation with a budget constraint for computing independent tasks in the cloud. *2010 IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom)*, páginas 319–328, 2010.
- [12] SUN, D., CHANG, G., WANG, C., XIONG, Y. y WANG, X. Efficient nash equilibrium based cloud resource allocation by using a continuous double auction. En *Computer Design and Applications (ICCD), 2010 International Conference on*, vol. 1, páginas V1–94–V1–99. 2010.