

SIMULADOR PARA ASIGNACIÓN DE RECURSOS EN NUBES BASADAS EN TEORÍA DE JUEGOS Y SISTEMAS DE REPUTACIÓN

Jefferson Valdez ⁽¹⁾, Juan Romero ⁽²⁾, Carlos Mera ⁽³⁾
Facultad de Ingeniería en Electricidad y Computación ⁽¹⁾
Escuela Superior Politécnica del Litoral (ESPOL)
Campus Gustavo Galindo, Km 30.5 Vía Perimetral
Apartado 09-01-5863. Guayaquil-Ecuador
wvaldez@espol.edu.ec ⁽¹⁾, jumarome@espol.edu.ec ⁽²⁾, cjmera@espol.edu.ec ⁽³⁾

Resumen

El presente trabajo tiene como objetivo diseñar un modelo basado en teoría de juegos y mecanismos de reputación, que ayude a mejorar la asignación de recursos en Cloud Computing y permita minimizar las violaciones SLA incluyendo la reputación de un proveedor Cloud como un nuevo parámetro en el proceso de negociación entre cliente y proveedor; para lo cual se modeló este proceso como un juego no cooperativo, en el que los proveedores Cloud compiten entre sí por la asignación de recursos de una tarea requerida por un usuario Cloud, y postulan sus ofertas en orden de menor a mayor reputación, las mismas que serán procesadas por un agente intermediario, el cual utilizará el mecanismo de reputación beta para seleccionar al proveedor que más se ajuste a los criterios de selección del usuario Cloud. Para realizar la etapa de experimentación, se implementará un simulador sobre el framework CloudSim, el mismo que se desarrollará en un ambiente web, utilizando el framework JSF y la librería de componentes PrimeFaces para mostrar los resultados gráficamente y facilitar la creación de interfaces de usuario.

Palabras Claves: Teoría de Juegos, reputación, SLA, Cloud Computing, asignación de recursos, Cloudsim

Abstract

The present work aims to design a model based on fundamental game theory and reputation systems concepts that help to improve resource allocation mechanisms in Cloud Computing and significantly reduces SLA violations, by the inclusion of Cloud provider reputation as a new parameter in the negotiation process between client and provider; therefore this process was modeled as a non-cooperative game, in which Cloud providers compete for resource allocation of cloud user's tasks, and bid their offers in order from lowest to highest reputation, these offers will be processed by an intermediary agent, who will use the beta reputation mechanism in order to select the cloud provider that best suits the user's selection criteria. In order to perform the experiments, a cloudsim-based simulator will be developed in a web environment using the JSF framework and PrimeFaces library components to display the results graphically and simplify the user interfaces creation.

Keywords: Game Theory, reputation, SLA, Cloud Computing, resource allocation, Cloudsim

1. Introducción

Cloud Computing es un modelo computacional que propone una nueva forma de gestionar servicios de tecnologías de información y aplicaciones. Este modelo se caracteriza por ofrecer a los usuarios finales un conjunto de servicios bien definido, para responder a las necesidades de una empresa o negocio, reduciendo así costos en tiempo y contratación de personal especializado para realizar tareas cotidianas que a menudo se presentan en el mundo de las tecnologías de información. Una de las características

más importantes de esta tecnología es la escalabilidad y el pago por el uso del servicio, ya que está diseñada para ajustarse a medida que las necesidades de una organización aumentan o requieren de más poder computacional, así pues el usuario final solamente paga por el uso de servicios y se olvida de otros costos que normalmente serían considerados si no se contaran con estos beneficios y se trabajara de la forma tradicional.

2. Sistemas de Reputación

Los sistemas de reputación, como sugiere [1], ayudan a construir confianza en las comunidades y generalmente son usados cuando los usuarios que interactúan realizando transacciones, no tienen conocimiento uno del otro, así pues basándose en conocimiento previo forman una opinión que podría ayudar a otros usuarios a tomar una decisión.

La reputación, como define [1] es “la información recolectada y procesada sobre el comportamiento de una entidad según la experiencia de los demás”, los sistemas de reputación precisamente van llevando registros históricos sobre el comportamiento de las entidades involucradas, para luego procesarlos y transformarlos en información útil para el usuario. Acorde con [1] la arquitectura de un sistema de reputación está constituida por varios componentes y actores, tal como se muestra en la **figura 1**

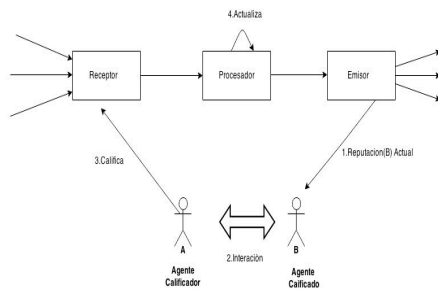


Figura 1. Arquitectura de un Sistema de Reputación

2.1 Métricas en sistemas de reputación

Un sistema de reputación puede ser descrito por sus métricas, que representan la forma en la cual se comparan diferentes agentes para medir su reputación. En [1] se clasifica a los sistemas de reputación según sus métricas de la siguiente manera:

Sistemas Acumulativos

El sistema recibe y acumula todas las calificaciones en orden para calcular la reputación global de un agente.

Sistemas Promedios

El sistema calcula la reputación de un agente como el promedio de todas las calificaciones que éste ha obtenido, lo cual se fundamenta en el hecho de que un agente se comportará de la misma manera la mayor parte del tiempo.

Sistemas Blurred

El sistema calcula la reputación de un agente como la suma de medias ponderadas de todas las calificaciones, con la diferencia de que las nuevas calificaciones tienen más influencia para la reputación actual, esto se basa en el hecho de que los agentes pueden cambiar su comportamiento durante su existencia.

Sistemas Only Last

El sistema calcula la reputación de un agente a partir de la última calificación que éste ha obtenido, fundamentado en el hecho de que un agente se comporte de la misma forma que la última vez.

Sistemas Eigen Trust

El sistema combina los valores de reputación local de un agente de manera iterativa para obtener una reputación global, la reputación local se define por un conjunto de agentes que pueden tener diferentes reputaciones en cada sistema de reputación local, pero solo un único valor en un sistema de reputación global.

Sistemas Adaptativos

El sistema se comporta de manera distinta en función de la reputación actual del agente; por ejemplo, si la reputación de un agente es baja, esto hace que una calificación positiva cause un alto incremento en la reputación, mientras que si la reputación es alta, la misma calificación produce únicamente un incremento pequeño en la reputación.

Sistemas Beta

El sistema trata de predecir el comportamiento futuro de un agente usando estadística. Esta predicción toma en consideración el comportamiento pasado de un agente y la probabilidad de comportarse bien o mal en la siguiente transacción.

2.2 Teoría De Juegos

La teoría de juegos estudia “como los individuos racionales toman decisiones cuando estos son mutuamente interdependientes” [2], es decir que las acciones que tomen repercutirán en los demás individuos. En [2] también se menciona que para poder comprender mejor esta definición se requiere tener en claro tres conceptos claves, que son: individualismo, racionalidad e interdependencia mutua.

El individualismo se refiere al hecho de que los participantes de un juego siempre buscarán su beneficio personal como principal objetivo sin importar lo que suceda. La racionalidad como define [2] asume que los individuos siempre actuarán a favor de sus intereses personales, y no tomarán decisiones que vayan en contra de los mismos. Finalmente, la interdependencia mutua significa que en cualquier momento las acciones de un individuo en un juego, pueden estar por lo menos parcialmente determinadas o condicionadas por las acciones de otros jugadores. Acorde con [2] los juegos pueden ser cooperativos y no cooperativos. Como se muestra en la **figura 2** en un juego no cooperativo cada individuo actúa de manera aislada para buscar el máximo beneficio, lo cual no significa que no existirán acuerdos entre jugadores, esto se dará solamente si favorece los intereses personales de los individuos.

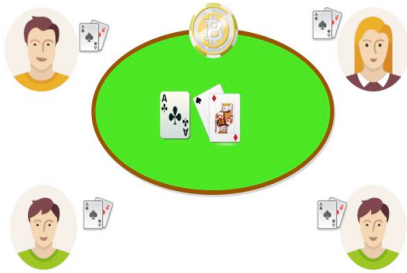


Figura 2. Representación de un juego no cooperativo

2.3 Service Level Agreement (SLA)

Un acuerdo de nivel de servicio (*SLA*), tal como sugiere [3], es un “documento que describe el nivel de servicio esperado por un cliente de un proveedor, definiendo las métricas en las que el servicio se medirá, y las soluciones o sanciones en caso de que estos niveles no se puedan lograr”.

El *SLA* cubre por lo general, una lista de los servicios que se entregarán, informes de seguimiento y rendimiento de recursos, la gestión de estrategias contra fallos, los deberes y responsabilidades de los clientes, la manera como se manejará la información confidencial, el tiempo de vigencia del contrato y los términos para el finiquito del mismo.

Dependiendo de la jerarquía de servicio (*SaaS*, *PaaS*, *IaaS*) a la que pertenezca, el proveedor deberá definir un "nivel objetivo de servicio", o a su vez un "nivel mínimo de servicio" que especifique las garantías del sistema, velocidad de conectividad, la disponibilidad, mantenimiento, funcionamiento y operación de los recursos; aceptando las penalizaciones a las que se debe someter en caso de incumplimiento. La **figura 3** ilustra gráficamente lo descrito anteriormente:

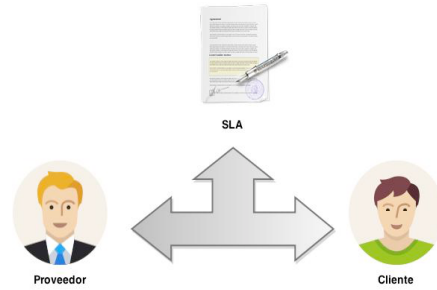


Figura 3. Service Level Agreement

3. Cloudsim

CloudSim es una herramienta de código abierto implementada en Java que permite simular escenarios que ocurren en la nube, a través de la simulación del hardware y software de un proveedor Cloud, el modelamiento de centros de datos, establecimiento de políticas de asignación de recursos, administración de máquinas virtuales y control de ancho de banda, son algunas de las configuraciones que pueden simularse, con la ventaja que las simulaciones se pueden replicar de manera sencilla, así, los ingenieros de TI pueden realizar pruebas robustas de la futura implementación sin que esto represente un costo adicional y controlando todas las variables necesarias. [4]. Entre las características que *CloudSim* ofrece, se pueden encontrar clases para el modelamiento de infraestructuras de centro de datos, administración de máquinas virtuales, políticas de asignación de recursos, etc. Esto hace que sea más fácil realizar un análisis cruzado de costo y beneficio tanto para cliente que busca la mejor oferta como para el proveedor que desea aumentar sus utilidades.

3.1 Arquitectura Cloudsim

La capa de simulación de CloudSim provee soporte para modelar y simular entornos de Datacenters basados en Cloud Computing, incluyendo interfaces de administración dedicadas para máquinas virtuales, almacenamiento y ancho de banda, esto podría ser de gran utilidad para los Proveedores Cloud que deseen probar nuevas estrategias de asignación de recursos y decidir así hasta qué punto son viables de implementar en un ambiente real.

4. Diseño Del Modelo

El experimento tiene como objetivo diseñar un modelo basado en teoría de juegos, el cual haciendo uso de un sistema de reputación, ayude a mejorar la asignación de recursos en Cloud Computing y reduzca las violaciones SLA por parte de los proveedores de Servicio.

4.1 Descripción del modelo

El modelo consiste en un juego no cooperativo en el cual los proveedores Cloud compiten por la asignación de recursos y envían sus ofertas en orden de menor a mayor reputación hacia un agente intermediario, el cual usará un sistema de reputación Beta [1] que predecirá el futuro comportamiento del proveedor en la siguiente transacción, para que así el precio no sea el único factor que haga que éste sea escogido. Los actores y componentes que han sido considerados para el modelo se muestran en la **figura 4** y se describen a continuación:

CRAG: Cloud Resource Allocation Game es un juego no cooperativo en el cual los proveedores Cloud compiten por la asignación de recursos, a partir de un trabajo enviado por el Usuario Cloud.

Usuario Cloud: El Usuario Cloud envía un trabajo al CRAG, en el cual especifica requerimientos específicos de recursos, el precio máximo que está dispuesto a pagar, y los parámetros de nivel de servicio en un SLA.

Trabajo: Representa una tarea que será enviada al juego, y está definida en millones de instrucciones por segundo

SLA: Componente que contiene especificaciones formales sobre los parámetros de nivel de servicio tales como: disponibilidad, confiabilidad, etc.

Proveedor Cloud: Es el dueño de los recursos Cloud y compite por la asignación de recursos en su infraestructura.

Agente Intermediario: Es el intermediario entre el Usuario Cloud y el Proveedor Cloud, el cual se encarga de encontrar el precio más económico con el menor riesgo de violaciones SLA para el Usuario Cloud, y recibe una comisión por parte del Proveedor por cada trabajo ejecutado exitosamente.

Recurso Cloud: Son las infraestructuras de hardware

que poseen cada uno de los proveedores Cloud, en este modelo los recursos serán los hosts que están dentro de un datacenter y poseen máquinas virtuales con una capacidad específica de cómputo medida en millones de instrucciones por segundo.

Mecanismo de Reputación Beta: Recopila las calificaciones que ha recibido un Proveedor Cloud para predecir de manera estadística el comportamiento del mismo en la siguiente transacción.

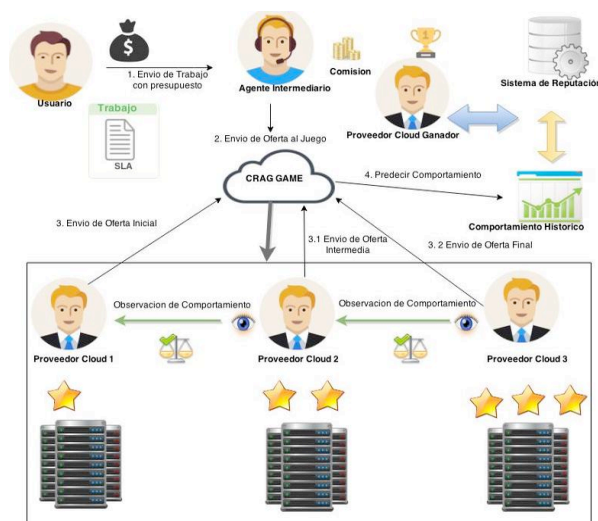


Figura 4. Representación del modelo

El juego inicia cuando el Usuario Cloud envía un trabajo al CRAG, el mismo que tiene adjunto un SLA y un valor máximo a pagar por la asignación de recursos, estos 2 parámetros son llamados de inicialización.

Adicionalmente, el Usuario Cloud define un nivel de importancia para la reputación, este parámetro junto con los parámetros de inicialización son procesados por el Agente Intermediario, de los cuales solamente el trabajo expresado en MIPS (Millones de Instrucciones Por Segundos) es dado a conocer a los Proveedores Cloud que estén compitiendo.

Los Proveedores Cloud harán sus ofertas en orden de menor a mayor reputación, de esta forma los que tienen mayor reputación pueden ajustar su nivel de ganancia acorde a las ofertas que hayan sido realizadas anteriormente, para así aumentar las posibilidades de ganar el juego. Este ajuste de ganancias que realizan los Proveedores Cloud mientras el juego se desarrolla, forma parte del modelo de teoría de juegos propuesto, el mismo que se clasifica dentro de teoría de juegos dinámica.

El *CRAG* es un juego dinámico debido a que los jugadores observan las acciones de otros jugadores, antes de tomar una decisión, con el objetivo de jugar su estrategia más óptima acorde al entorno de ese momento en particular. Los Proveedores Cloud establecen inicialmente un costo de producción por MIPS, el mismo que será incrementado en un porcentaje para convertirse en el precio de mercado. Los porcentajes de ganancia mínima y deseada, son definidos inicialmente pero estos pueden variar en función de la mejor oferta que se haya postulado en el juego.

Se dirá que un proveedor postulará su precio mínimo si aplica el porcentaje de ganancia mínimo, y que postulará su precio deseado si aplica el porcentaje de ganancia deseado.

Las estrategias que jugarán los proveedores Cloud se listan a continuación:

1. Aumentar el porcentaje de ganancia deseada en relación a la mejor oferta, solo si al hacerlo, el precio deseado se mantiene por debajo de la mejor oferta.
2. Aumentar el porcentaje de ganancia mínimo en relación a la mejor oferta solo si el precio deseado es mayor a la mejor oferta y si al hacerlo, el precio mínimo se mantiene menor a la mejor oferta.
3. Mantener el porcentaje de ganancia mínimo si no hay forma de ofrecer un precio menor al de la mejor oferta. Una vez que todos los proveedores han enviado sus ofertas al *CRAG*, este se encarga de procesarlas y ajustarlas acorde a los parámetros de configuración recibidos inicialmente, para ello utilizará el mecanismo de reputación Beta [9] para predecir el comportamiento del proveedor en la siguiente transacción.
4. Finalmente, el agente intermediario selecciona al Proveedor Cloud que resulte ganador del *CRAG*. La asignación de recursos se realiza en la infraestructura del ganador y el agente intermediario recibe una comisión por parte del Proveedor Cloud, el cual consiste en el 15% del precio que el usuario cloud ha pagado por la asignación de recursos.

5. Diseño Del Simulador

La **figura 5** muestra las clases principales del simulador y sus relaciones utilizadas en la implementación del mismo, las clases que fueron implementadas para el simulador se muestran de color celeste, mientras que las incorporadas en *CloudSim* se muestran de color verde

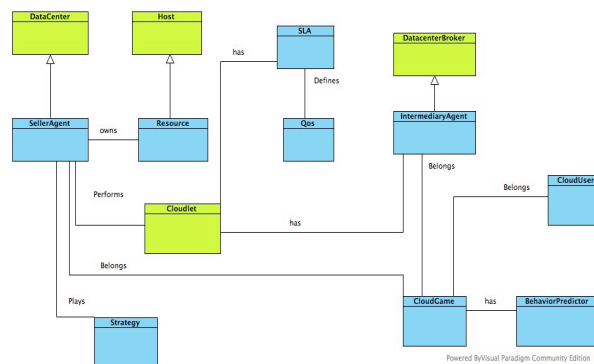


Figura 5. Diagrama de clases del simulador

Las clases más significativas del simulador son descritas a continuación:

DataCenter: Esta clase modela los principales niveles de servicio a nivel de infraestructura, ofrecidos por un servidor en un ambiente de Cloud Computing.

Host: Esta clase modela un servicio físico en un datacenter Cloud, contiene una cantidad de memoria, almacenamiento y una lista de elementos de procesamiento.

Cloudlet: Esta clase modela los servicios de aplicaciones basadas en Cloud Computing, que son comúnmente desplegadas en los datacenters.

DataCenterBroker: Esta clase modela un agente intermediario, que es responsable de mediar entre los usuarios cloud y los proveedores de servicios, para poder determinar los proveedores más convenientes que cumplan con los parámetros de calidad de servicio definidos por el usuario.

SellerAgent: Esta clase hereda las funcionalidades de la clase Datacenter e incluye nuevos atributos y métodos relacionados al simulador.

IntermediaryAgent: Esta clase hereda de la clase DatacenterBroker e incluye operaciones para regular el proceso de negociación entre el usuario y el proveedor de servicios cloud.

BehaviorPredictor: Esta clase utiliza el mecanismo de reputación Beta para predecir el comportamiento futuro de un proveedor Cloud.

CloudGame: Esta clase modela la funcionalidad principal del simulador, y procesa todos los parámetros previos a la simulación de un escenario.

6. Experimentación

Para efectuar las simulaciones se estableció un escenario inicial, sobre el cual se ejecutó 5 rondas de simulaciones, las cuales estuvieron compuestas de 10 iteraciones cada una, cabe recalcar que las iteraciones no son repeticiones, debido a que los eventos que ocurran en cada una de ellas influirán en la siguiente iteración. El escenario inicial se muestra en la **tabla 1**, las columnas RI, GM y GD representan: reputación inicial, porcentaje de ganancia mínima y porcentaje de ganancia máxima respectivamente.

Tabla 1. Escenario inicial

PROVEEDOR	RI	COSTO	GM	GD
A	3	\$1	0,2	0,4
B	5	\$1,3	0,3	0,5
C	7	\$1,6	0,3	0,5
D	9	\$1,9	0,4	0,5

Para cada una de las rondas, la importancia de reputación fue aumentándose progresivamente de 2 en 2, comenzando en 2 y terminando en 10, con esto se esperaría que a medida que la importancia de reputación aumente, el número de violaciones SLA disminuyan

7. Resultados

Los resultados de la experimentación se muestran a continuación:

Tabla 2. Resultados de simulación con importancia de reputación 2

Proveedor	IMPORTANCIA DE REPUTACIÓN 2										
A	2	2	2	2	2	2	2	2	2	2	2
B	2	2	2	1	2	2	2	2	2	2	0
C	2	2	1	2	1	1	0	1	1	1	1
D	1	1	1	1	1	0	0	0	0	0	0
TOTAL	7	7	6	6	6	5	4	5	5	3	



Figura 6. Violaciones SLA para la simulación 1

Tabla 3. Resultados de simulación con importancia de reputación 4

Proveedor	IMPORTANCIA DE REPUTACIÓN 4									
A	2	2	1	2	1	0	1	2	1	1
B	1	2	2	1	2	1	1	1	1	0
C	2	1	2	1	0	1	1	1	0	0
D	1	1	1	1	1	1	1	0	0	0
TOTAL	6	6	6	5	4	3	4	4	2	1



Figura 7. Violaciones SLA para la simulación 2

Tabla 4. Resultados de simulación con importancia de reputación 6

Proveedor	IMPORTANCIA DE REPUTACIÓN 6									
A	1	1	1	1	1	0	1	1	0	1
B	1	1	1	0	1	1	1	1	1	0
C	1	1	1	1	0	1	1	0	1	0
D	0	1	0	1	0	1	0	0	0	0
TOTAL	3	4	3	3	2	3	3	2	2	1



Figura 8. Violaciones SLA para la simulación 3

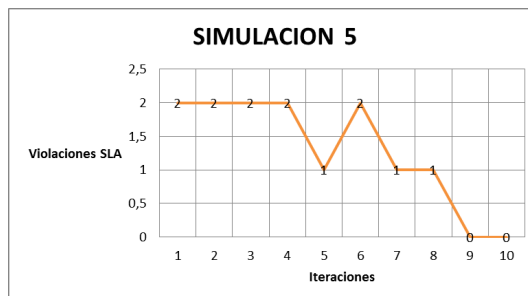


Figura 10. Violaciones SLA para la simulación 5

Tabla 5. Resultados con importancia de reputación 8

Proveedor	IMPORTANCIA DE REPUTACIÓN 8									
A	1	1	1	1	1	1	1	1	1	0
B	1	1	1	0	1	1	0	0	0	0
C	0	1	1	1	0	0	0	0	0	0
D	1	1	0	1	0	1	0	0	0	0
TOTAL	3	4	3	3	2	3	3	2	2	1

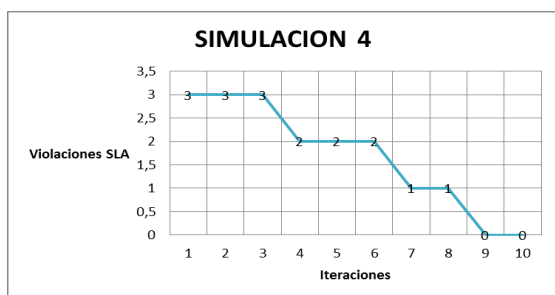


Figura 9. Violaciones SLA para la simulación 4

Tabla 6. Resultados con importancia de reputación 10

Proveedor	IMPORTANCIA DE REPUTACIÓN 10									
A	1	1	0	1	1	1	1	1	0	0
B	1	0	1	0	0	1	0	0	0	0
C	0	0	1	1	0	0	0	0	0	0
D	0	1	0	0	0	0	0	0	0	0
TOTAL	2	2	2	2	1	2	1	1	0	0

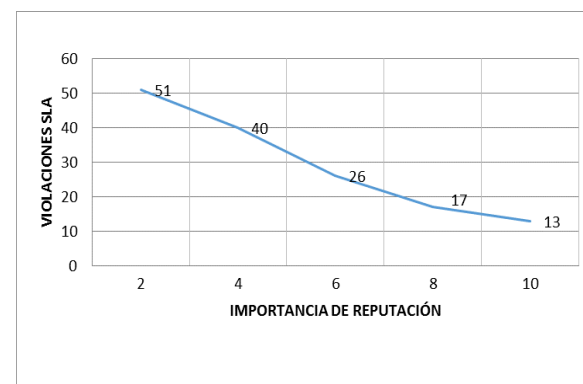


Figura 11. Violaciones SLA por simulación

Para cada una de las rondas de simulaciones realizadas, se obtuvo el número de violaciones SLA por cada proveedor en cada iteración. En todas las simulaciones, se pudo observar que el mayor número de violaciones es cometido por los proveedores con más baja reputación.

Finalmente en la figura 6 se muestra el número total de violaciones acumuladas que han cometido los proveedores Cloud por cada ronda de simulación, y se puede notar claramente que al aumentar el nivel de importancia de reputación, el número de violaciones SLA tiende a reducirse considerablemente.

8. Conclusiones

Finalmente se puede concluir que un modelo que toma en cuenta la reputación de los proveedores para establecer un mecanismo de negociación entre cliente y proveedor, siempre será de beneficio para el usuario cloud ya que, no solamente se puede guiar por el precio, sino también por el nivel de calidad de servicio.

La inclusión de un modelo que toma en cuenta la reputación de los proveedores Cloud como criterio de selección para realizar tareas requeridas por un usuario Cloud, el número de violaciones SLA se reduce en

promedio, a medida que el nivel del parámetro importancia de reputación es incrementado.

Una limitación del modelo propuesto reside en el sistema de reputación Beta, debido a que no se cuenta con un mecanismo que pueda detectar cambios en el comportamiento de un proveedor, se recomienda añadir al modelo, un factor de olvido, en el cual las calificaciones más antiguas que reciban los proveedores Cloud tengan menos peso que las más recientes, de esta manera se deja abierta la posibilidad de que el comportamiento de un proveedor pueda cambiar a lo largo del tiempo y con ello mejorar su reputación.

9. Referencias

- [1] Schlosser, A., Voss, M., Bruckner, L., “Comparing and Evaluating Metrics for Reputation Systems by Simulation”, [En línea].
Disponible en:
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.142.2347&rep=rep1&type=pdf>
- [2] Romp, G., “Game Theory: Introduction and Application”, Oxford University Press, 1997
- [3] Lynn, G., y Lauren, P., “SLA Definitions and Solutions”, [En línea].
Disponible en:
<http://www.cio.com/article/2438284/outsourcing/sla-definitions-and-solutions.html>
- [4] Calheiros, R., Ranjan, R., y De Rose, C., “CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,” [En línea].
Disponible en:
<http://www.buyya.com/papers/CloudSim2010.pdf>
- [5] National Institute of Standards and Technology, The NIST Definition of Cloud Computing,
Disponible en:
<http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>
- [6] Antonopoulos, N., Gillam, L., Cloud Computing: Principles, Systems and Applications, Springer, 2010.
- [7] Mera, C., Simulation Tool for Market-Based Cloud Resource Allocation,
Disponible en:
<http://repositorio.educacionsuperior.gob.ec/bitstream/28000/277/1/T-SENESCYT-0047.pdf>, fecha de consulta 15 de marzo 2015
- [8] Guiyi, W., Athanasios, V., Vasilakos, Y., Yao, Z., Naixue, X., A game-theoretic method of fair resource allocation for cloud computing services,
Disponible en:
http://140.123.102.14:8080/reportSys/file/paper/cktsung_g/cktsung_39_paper.pdf, fecha de consulta 15 marzo 2015
- [9] Rodríguez, F., Teoría de juegos: análisis matemático de conflictos,
Disponible en:
<http://imarrero.webs.ull.es/sctm05/modulo1lp/5/ffernandez.pdf>, fecha de consulta Julio 2014.
- [10] Farmer, R., Glass, B., Building Web reputation systems, O'Reilly 1st Ed, 2010.
- [11] Guiyi, W., Athanasios, V., Vasilakos, Y., Yao, Z., Naixue, X., A game-theoretic method of fair resource allocation for cloud computing services,
Disponible en:
http://140.123.102.14:8080/reportSys/file/paper/cktsung_g/cktsung_39_paper.pdf, fecha de consulta 15 marzo 2015