

Modelado y simulación de arquitecturas de Cloud Computing con CloudSim: comunicación entre entidades.

Rosatto Daniel¹, Bond Román¹, Belizán Maximiliano¹, Morales Martín^{1,2}, Encinas Diego^{1,3}

¹Instituto de Ingeniería y Agronomía - Universidad Nacional Arturo Jauretche

²Unidad CodApli - Facultad Regional La Plata - UTN

³Instituto de Investigación en Informática LIDI (III-LIDI) - Facultad de Informática - UNLP

danielrosatto@gmail.com, roman.alejandro.b@gmail.com, maximiliano.h.belizan@gmail.com,
martin.morales@unaj.edu.ar, dencinas@unaj.edu.ar

Resumen. Las diferentes herramientas de simulación para las distintas áreas del conocimiento son excelentes sistemas proveedores de posibles comportamientos en cuanto a obtener conclusiones previas durante el diseño. En el caso de los sistemas de Cloud Computing, la diversidad de herramientas de simulación es escasa e insuficiente, comparando con otros sistemas de simulación como por ejemplo redes de computadoras. En este trabajo se abordará el desarrollo obtenido como una alternativa válida para la implementación de entidades de simulación capaces de soportar intercambios de datos simulados en el contexto de un framework de simulación de arquitecturas de Cloud Computing.

Palabras clave: Cloud Computing, Redes, Simulación.

1 Introducción.

Las ventajas de los sistemas Cloud se combinan con la centralización de la autenticación, la seguridad en el acceso y el licenciamiento de los sistemas virtualizados. Los grandes proveedores de servicios Cloud, desarrollaron productos comerciales para cubrir las necesidades TI de pequeñas, medianas y grandes empresas. En este marco, es posible desarrollar conceptos académicos pero siempre hay restricciones que se vinculan con las características del despliegue, o restricciones relacionadas con los servicios que brinda el proveedor de Cloud. Algunas son: los códigos fuente de los servidores Cloud generalmente no se publican, no es posible acceder a un paquete compilado, o nunca se sabe fehacientemente en donde se encuentran los datos o información.

Al comenzar con el desarrollo de un gran despliegue de un sistema Cloud, los posibles riesgos a abordar son los costos, la incertidumbre de la implementación del paradigma, el nivel de confianza y seguridad que se le proveerá al cliente, según lo expresado en el acuerdo del nivel de servicio. Una buena forma de previsualizar los

resultados es mediante la utilización de sistemas de simulación, configurando los posibles escenarios que reflejen situaciones reales y analizando los datos de salida para determinar si es viable o no realizar tal despliegue.

La herramienta de simulación utilizada es el framework CloudSim, que mediante entidades (datacenters, máquinas virtuales, etc), permite reflejar diferentes escenarios de infraestructura Cloud. Debido a las necesidades de este trabajo y ya que el core de CloudSim permite el agregado de nuevas funcionalidades, se desarrolla una expansión de CloudSim. En el siguiente apartado se explican los criterios de elección del framework CloudSim para realizar el aporte.

1.1 Simulación de arquitecturas de Cloud Computing.

CloudSim [1] [2] es un framework de simulación generalizado y extensible que permite el modelado, la simulación y la experimentación de diferentes infraestructuras y servicios de aplicaciones de Cloud Computing. Un ejemplo de utilización es la simulación de muchos centros de datos.

Su arquitectura consiste en entidades específicas que se representan como clases Java que pueden ser heredadas o variadas para simular experimentos. Estas clases representan centros de datos, hosts físicos, máquinas virtuales, servicios a ejecutar en los centros de datos, servicios en la nube de usuarios, redes internas, centro de datos y consumo de energía de los hosts físicos y elementos de los centros de datos. Además, CloudSim soporta la inserción dinámica de los elementos de simulación y proporciona aplicaciones de paso de mensajes y la topología de la red del centro de datos.

Una definición importante de CloudSim es la de entidad. Una entidad es una instancia de un componente, que es una clase o un conjunto de clases que representan un modelo CloudSim (data center, host). El motor de simulación es capaz de simular el tiempo de ejecución de las aplicaciones ingresadas como Cloudlets con información básica [3].

La versatilidad de CloudSim es la principal ventaja del sistema. La integración de nuevos parámetros y conceptos de la simulación es implementada desde abstracciones preestablecidas convenientemente en este trabajo. Las abstracciones principales son: SimEvent [4], SimEntity [4], DataCenterCharacteristics y Virtual Machine (Vm). En cada una de estas instancias es posible establecer el detalle del comportamiento y el estado de la simulación teniendo en cuenta el alcance del modelado que se planifica.

Se propone una simulación que provea métricas de entrada/salida de servicios Cloud simulados tales como SAAS, PAAS o IAAS desarrollada con objetos que heredan de SimEvent y de SimEntity pero con sus métodos y estados únicos en cada simulación. Las denominaciones de los servicios SAAS, PAAS e IAAS hacen referencia al objetivo del sistema Cloud, es decir cuál es el servicio que provee.

- Un sistema SAAS, (Software as a service) brinda servicios de software a través del acceso Cloud, como por ejemplo el reproductor de videos online de Youtube.
- Un sistema PAAS (Platform as a service) brinda una plataforma de hardware virtualizado a través del acceso Cloud, como por ejemplo los storages de archivos como SugarSync.

- Un sistema IAAS (Infrastructure as a service) brinda una infraestructura de hardware y software completa accesible a través de acceso Cloud, como por ejemplo, los servidores virtuales de Iplan.

Los grandes proveedores de servicios Cloud brindan los tres paradigmas juntos, de tal manera que proporcionan servicios integrales.

1.2 Simulación de procesamiento de datos.

CloudSim establece que el entorno de simulación es un sistema de Data Centers con la administración de un broker que asigna los recursos disponibles. En el modelo de simulación de CloudSim, el objetivo es procesar todas las instancias de Cloudlets, que se definen al inicio de la simulación.

Las Cloudlets son definiciones de unidades de información que cualquier Dashboard de un sistema Cloud genera por la interacción de los distintos clientes. En el modelo de CloudSim, las Cloudlets contienen los requisitos de memoria necesaria, de forma que el broker asigne a una Vm capaz de procesarlo, según su algoritmo de asignación.

2 Desarrollo.

Se logró crear un desarrollo basado en las clases SimEntity y SimEvent de CloudSim, utilizando metodologías de expansión de trabajos previos [5] y [6]. Estas clases no están vinculadas con los Cloudlets porque el objetivo del modelado del ambiente de simulación de CloudSim no es procesar algoritmos de entrada. Sólo obtener la cantidad de Cloudlets procesados por cada data center y cuál fue su costo temporal. En éste caso se establecen las modificaciones en el ámbito de CloudSim para configurar a la memoria Ram de cada Vm (Virtual Machine) como una entidad de simulación que procesa eventos de simulación. Es la condición inicial de la simulación para observar y medir procesos distribuidos en cada una de las Vm del broker. Los datos en las celdas de cada SimEntity Ram, (que se ha denominado RamEntity) serían sincronizados por los eventos de simulación autogenerados por la lógica del nuevo modelo. La medición de los procesos se realizaría en el contexto de un Benchmark precargado en el modelo.

En lo referido a Cloudsim, los servicios Cloud que implementa son capaces de procesar las cloudlets que definen los creadores de CloudSim. En este caso la opción es evaluar los resultados obtenidos en el Log de CloudSim al finalizar el proceso sintético de Benchmark, que interactúa con el Sistema de Archivos sintético de CloudSim. Se explica en el siguiente apartado los llamados y las delegaciones del control entre cada uno de los actores de la simulación.

2.1 Expansión de entidades.

Para crear una nueva entidad cloudsim se debe proceder de manera similar a los cloudsim-examples pero adecuándose a las necesidades y sin modificar los archivos originales de cloudsim.

Se incorporaron las siguientes entidades: primero, una RamEntity como componente de las máquinas virtuales en reemplazo de la variable ram.

En segundo lugar, se cambió la clase Vm por SyntheticVm con una composición de RamEntity.

Los procesos sintéticos surgen de la necesidad de respuesta de la simulación ante factores internos. La delegación de la ejecución de las líneas de procesos sintéticos logra que la respuesta de su algoritmo sea obtenida por los recursos simulados de CloudSim independiente del hardware real que está ejecutándose. En la Figura 1 se observa que toda la simulación depende exclusivamente de los recursos que provee la capa de simulación, debiendo reportar cada paso a la capa de control.

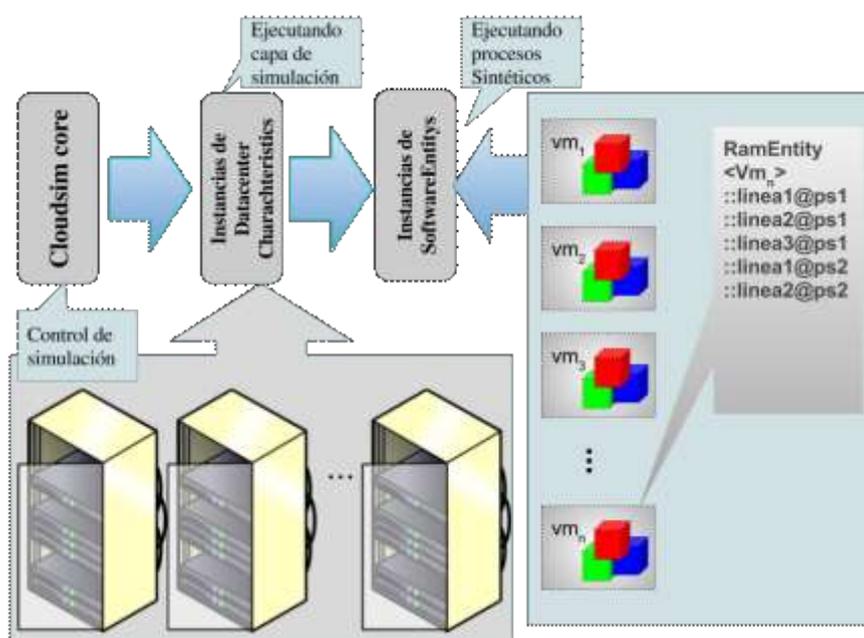


Figura 1: Diagrama de expansión de entidades de CloudSim

Se exponen a continuación el método de desarrollo de la entidad RamEntity, que proporciona el soporte para mantener la simulación de celdas accesibles mediante eventos de simulación.

Estudiando detalles de la clase abstracta SimEvent fue posible definir los requerimientos de software que la nueva entidad debe cumplir. Los requisitos son adecuados para la implementación de RamEntity a partir del cumplimiento de sus restricciones de software. Es por eso que se generan test JUnit RamEntity_test basado en cada caso de los requisitos de los métodos heredados de la superclase SimEntity.

Los Unit test producen una implementación de RamEntity que verifican el cumplimiento de los requerimientos de la superclase con el desarrollo aportado.

En la figura 2 se detallan las relaciones entre las nuevas entidades, el core del simulador y los eventos. Lo interesante en este modelo es que cada celda de RamEntity soporta la asignación de un Object de java. Es decir que se conserva la versatilidad original de CloudSim.

El evento inicial debe ser configurado antes de hacer un start simulation en el tiempo que se asigna en el método send del core de CloudSim. El evento al estar destinado al id (identificador) de la RamEntity2 y la fuente debe ser el id de RamEntity2. Será procesado por la simulación con la correspondiente etiqueta del evento, que en éste caso es una Ram_Access_Read. La versatilidad del simulador se refleja en la posibilidad de que RamEntity1 tiene la capacidad de generar un nuevo evento como respuesta.

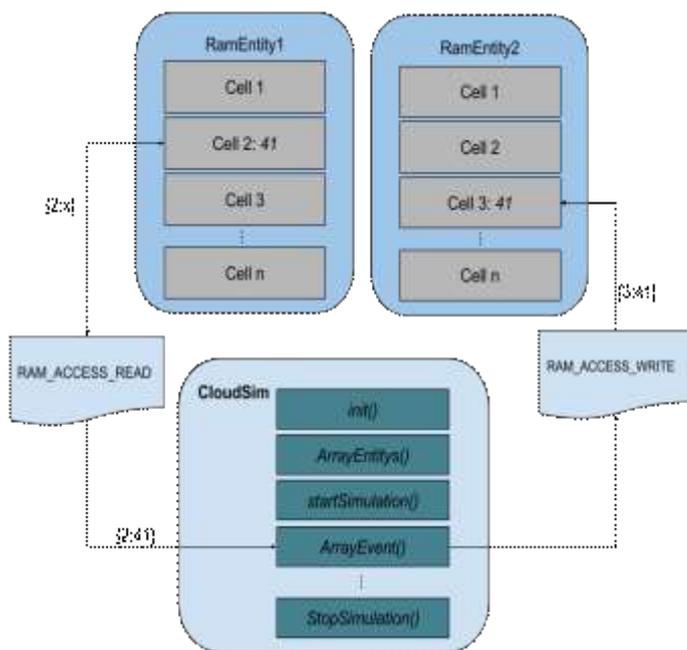


Figura 2: Diagrama de transferencia de información entre celdas RamEntity

Tabla 1: Salida LOG de la simulación de RamEntity.

```

Starting BenchmarkingCloudSimExample1...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entity >vm1&VRAM< : Cells init...
Entity >vm2&VRAM< : Cells init...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: Creation of VM #0 failed in Datacenter #2
0.1: Broker: Sending cloudlet 0 to VM #0
Entity >vm1&VRAM< :59.0 SUCCESS ACCESS READ: -1
Entity >vm1&VRAM< :79.0 SUCCESS ACCESS WRITE: -1 <- vm1&VRAM[23]=4
Entity >vm1&VRAM< :99.0 SUCCESS ACCESS READ: 4
Entity >vm2&VRAM< :101.0 SUCCESS ACCESS WRITE: 4 <- vm2&VRAM[23]=4
Entity >vm2&VRAM< :109.0 SUCCESS ACCESS READ: 4
Entity >vm1&VRAM< :: ENULL Ram event type, time: 118.91 ,data: -1
Entity >vm1&VRAM< :: ENULL Ram event type, time: 198.91 ,data: 4
Entity >vm2&VRAM< :: ENULL Ram event type, time: 218.91 ,data: 4
400.1: Broker: Cloudlet 0 received
400.1: Broker: All Cloudlets executed. Finishing...
400.1: Broker: Destroying VM #0
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Entity >vm1&VRAM< :Shutdown ...
Entity >vm2&VRAM< :Shutdown ...
Simulation completed.
Simulation completed.

===== OUTPUT =====
Cloudlet ID   STATUS   Data center ID   VM ID   Time   Start Time   Finish
Time
0             SUCCESS   2                0       400    0,1          400,1
Benchmarking CloudSim Example1 finished!

```

Luego, en el segundo resaltado cian, el evento que interactúa con vm1&VRAM produce una lectura de la celda 23, luego escribe 4 en la misma celda. Posteriormente se verifica en el tiempo 99 el valor 4 en la celda 23.

Seguidamente lo resaltado en amarillo verifica de manera similar la escritura de la celda 23 de vm2&VRAM. Los eventos ENULL posteriores representan a los eventos generados por las respuestas de las lecturas y escrituras de RamEntity. Los tiempos de las devoluciones ENULL son los sincronizados por el gestor de tiempos.



Figura 4: Comparación de los tiempos consumidos por cada respuesta de la RamEntity ante los eventos generados.

Se observa en la Figura 4 una comparación de los tiempos de respuesta obtenidos por las acciones de las SimEntity implementadas. En futuras versiones las respuestas de los tiempos se combinarán con un análisis estadístico para establecer las estimaciones de las variables temporales que se requieran.

Tabla 2: [9]

TIEMPO MEDIO DE INICIO DE MV		
Cloud	OS	Tiempo
EC2	Linux	96.9 segundos
EC2	Windows	810.2 segundos
Azure	WebRole	374.8 segundos
Azure	WorkerRole	406.2 segundos
Azure	VMRole	356.6 segundos

En la Tabla 2 se pueden observar los tiempos consumidos en ejecutar una máquina virtual, con distintos sistemas operativos, en los dos proveedores de Cloud Computing más conocidos. Con CloudSim el tiempo es mucho menor, para poder centrarse más en métricas de funcionamiento y de hardware simulado.

4 Conclusiones.

Se logró establecer un método de desarrollo adecuado para un framework de simulación de arquitecturas Cloud. El objetivo de lograr que las entidades de simulación soporten intercambios de datos fue cumplido exitosamente en la entidad RamEntity. Se destaca que el método es genérico para cualquier entidad que se quiera implementar a futuro de acuerdo a los objetivos del modelo.

La precisión del intercambio de los datos es susceptible al comportamiento implementado en el método que procesa los eventos. De manera que mientras más eventos se incluyan en la implementación de éste método, se incrementa la calidad de la transferencia de información sintética entre entidades.

Ampliar este framework ayuda a comprender no sólo los entornos Cloud Computing, sino que también se puede entender mejor este framework para poder seguir expandiéndolo y luego replicarlo en un espacio físico.

5 Trabajos futuros.

Esta línea de investigación es la base a una serie de extensiones de funcionalidades y características que continúan en la etapa de desarrollo. El modelo de simulación seguirá proporcionando la versatilidad original de CloudSim, y además se proporcionarán nuevos actores en el escenario de la simulación. Dichos actores serán entidades y eventos que se consideran relevantes según lo investigado en las experiencias actuales de desarrollo para complementar otras líneas de investigación [7] [8] obteniendo un modelo aplicable.

En los siguientes trabajos, los intercambios de datos en RamEntity darán la posibilidad de crear procesos simulados en una tabla de procesos, también simulada en cada SyntheticVm.

La tabla de procesos simulada estará incluida en la InitEntity que será una entidad administradora de procesos de cada SyntheticVm para dar una respuesta adecuada en cada VirtualMachine simulada.

6 Referencias

1. CloudSim - <http://www.cloudbus.org/cloudsim>. julio 2017
2. Rajkumar Buyya “CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms” Published online 24 August 2010 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/spe.995.
3. Kushang Parikh, Nagesh Hawanna, Haleema. P.K, Jayasubalakshmi.R and N.Ch.S.N.Iyengar. School of Computing Science and engineering Vellore Institute of Technology, Tamil Nadu, “Virtual Machine Allocation Policy in Cloud Computing Using CloudSim in Java.” 2015.
4. Fred Howell & Ross Mc Nab “SimJava: A Discrete Event Simulation Library For Java”. International Conference on Web-Based Modeling and Simulation. 1998.

5. Rodrigo N. Calheiros , Rajiv Ranjan, Anton Beloglazov, César A. F. De Rose and Rajkumar Weiwei Chen , Ewa Deelman - Information Sciences Institute-University of Southern California Marina del Rey, CA, USA - WorkflowSim: A Toolkit for Simulating Scientific Workflows in Distributed Environments. 2012
6. Hamza Ouarnoughi, Jalil Boukhobza, Frank Singhoff, Stephane Rubini, Erwann Kassis. “Considering I/O Processing in CloudSim for Performance and Energy Evaluation”. OpenStack Cloud Software: Open source software for building private and public clouds. © Springer International Publishing AG 2016 M. Tafer et al. (Eds.): ISC High Performance Workshops 2016, LNCS 9945, pp. 591–603, 2016. DOI: 10.1007/978-3-319-46079-6 40.
7. Zaccardi, G.; Galarza, B.; Encinas, D.; Morales, M. “Implementación de Cloud Computing utilizando OpenStack”. II Congreso Nacional de Ingeniería Informática y Sistemas de Información (CoNaIISI 2014). Ciudad de San Luis, Argentina.
8. Galarza, B.; Zaccardi, G.; Encinas, D.; Morales, M. “Análisis de despliegue de una IaaS utilizando Openstack”. XXI Congreso Argentino de Ciencias de la Computación (CACIC 2015). Ciudad de Junín, Argentina.
9. Ming Mao, Marty Humphrey “A Performance Study on the VM Startup Time in the Cloud”. IEEE 5th International Conference on Cloud Computing (CLOUD), 2012.