

Análisis de una plataforma de simulación para Cloud Computing. Un caso de estudio

Tomás Rosales¹, Julián Spinelli¹, Marcos Di Nardo¹, Román Bond², Daniel Rosatto², Diego Encinas^{1,2,3}, Fernando Romero^{1,3}

¹Facultad de Informática, Universidad Nacional de La Plata. Buenos Aires, Argentina

²Proyecto de Investigación SimHPC, Programa TICAPPS, Universidad Nacional Arturo Jauretche. Buenos Aires, Argentina

³Instituto de Investigación en Informática (III-LIDI). Facultad de Informática, Universidad Nacional de La Plata - Centro Asociado CIC. Buenos Aires, Argentina
tomasjoaquinrosales@gmail.com, julspni@gmail.com, mardinar@gmail.com, rbond@unaj.edu.ar, danielrosatto@gmail.com, dencinas@lidi.info.unlp.edu.ar, fromero@lidi.info.unlp.edu.ar

Resumen Se presenta una aproximación a la construcción de simuladores en entornos de Cloud Computing por medio de un framework específico. Con este fin, se hace uso de un sistema real desplegado en Amazon Web Services para obtener conjuntos de datos de salida relacionados a la ejecución de tareas en forma paralela. Y por medio de la plataforma de simulación CloudSim se desarrolla una simulación. Finalmente, se procede a la verificación y validación del simulador utilizando los datos de los sistemas físicos y simulados.

Palabras Claves: Simulación, Cloud Computing, CloudSim.

1. Introducción

Con el surgimiento de la idea de computación en nube (Cloud Computing) se han abierto nuevas posibilidades para todo el sector IT. Este paradigma está orientado a ofrecer recursos, aplicaciones y plataformas como servicios por Internet, con modelos de precios basados en el consumo utilizado [1]. Sus inicios se deben a proveedores de Internet de gran escala, que construyeron su propia infraestructura Cloud. La idea inicial, fue alquilar recursos ociosos de su propia infraestructura para usuarios que prefirieran utilizar recursos pagando exclusivamente bajo demanda.

Para poder maximizar los beneficios de este modelo computacional complejo, es necesario contar con herramientas que permitan comprender el funcionamiento del paradigma. Por ejemplo, es posible simular el procesamiento de una carga determinada por parte de una aplicación de comercio electrónico y estimar cuánto costaría su ejecución en diferentes proveedores de servicios de Cloud [2]. Las simulaciones, además, son esenciales para intentar prever el comportamiento de la aplicación y para estimar costos, antes de ubicar la aplicación en una determinada plataforma de operación.

En este trabajo se utiliza CloudSim para ejecutar simulaciones de escenarios en Cloud Computing. Cloudsim basa su funcionamiento en un conjunto de herramientas de simulación extensibles que permite el modelado y simulación de sistemas de computación en la nube. El objetivo del presente trabajo es configurar Cloudsim para reflejar las características ofrecidas por un conocido proveedor de servicios (AWS, Amazon Web Services) [3]. Luego se miden los tiempos de respuestas de ambos entornos para una tarea determinada y así obtener conclusiones generales que serán especialmente útiles al momento de determinar la configuración más conveniente para desplegar la aplicación.

2. CloudSim

CloudSim [4] es un framework que permite modelar y simular diferentes arquitecturas de Cloud Computing. Mediante esta herramienta es posible representar cada uno de los componentes intervinientes en la nube a través de clases de Java. A partir de dichas clases se modelan las diferentes entidades, como Datacenters, Brokers, Hosts, Máquinas Virtuales, Tareas (Cloudlets), Servicios y Redes, entre otras, cada una con parámetros propios que las caracterizan [5] como se puede ver en la *Figura 1*.

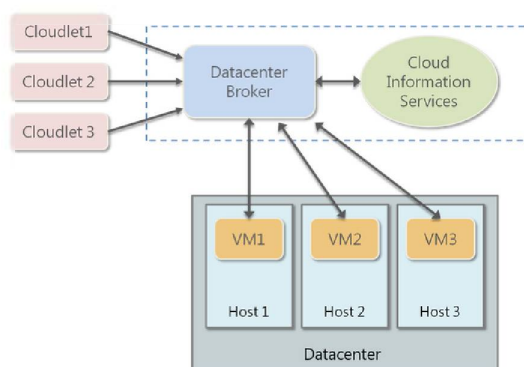


Figura 1. Principales entidades en CloudSim.

El escenario de simulación más básico a configurar y ejecutar, consiste en un Data Center con la administración de un broker que asigna los recursos disponibles [6]. El objetivo es procesar todas las instancias de Cloudlets.

2.1. Cloudlet

Esta clase representa las tareas que deben ejecutarse en la infraestructura de la nube para su procesamiento. [7]. Cada cloudlet tiene una sobrecarga de

instrucciones y transferencia de datos que debe realizar durante su ciclo de vida. En toda simulación de CloudSim los cloudlets interactúan con el broker y los datacenters con el objetivo que cada Virtual Machine (instancia) sea capaz de procesarlo, según su algoritmo de asignación. En el escenario de simulación que se propone para este trabajo, los parámetros de cada cloudlet son los que se modifican para poder reflejar la carga de trabajo de una tarea determinada.

```

1 // Fifth step: Create one Cloudlet
2 cloudletList = new ArrayList<Cloudlet>();
3
4 // Cloudlet properties
5 int id = 0;
6 long length = 400000;
7 long fileSize = 300;
8 long outputSize = 300;
9 UtilizationModel utilizationModel =
10     new UtilizationModelFull();
11 Cloudlet cloudlet = new Cloudlet(id, length, pesNumber,
12     fileSize, outputSize, utilizationModel,
13     utilizationModel, utilizationModel);

```

A continuación se describen cada uno de los parámetros que describen a un cloudlet:

1. id: identificador único. Importante para el motor de cloudsims, ya que lleva el control de qué cloudlets ejecutó y cuáles no.
2. length: el tamaño del cloudlet en MiB.
3. fileSize: el tamaño de archivo del cloudlet antes de enviarlo a un Datacenter para su procesamiento.
4. outputSize: el tamaño de archivo del cloudlet después de su ejecución en un Datacenter.
5. pesNumber: hace referencia a la unidad de procesamiento. Se define en términos de millones de instrucciones por segundo (MIPS).

Los últimos 3 parámetros hacen referencia al modelo de utilización de la cpu, la memoria ram y del ancho de banda (red), respectivamente. A su vez, las variables que hacen referencia al tamaño de archivo del cloudlet están en bytes.

3. AWS

Amazon Web Services (AWS) es una colección de servicios de computación que en conjunto forman una plataforma de computación en la nube, ofrecidas a través de Internet por Amazon.com. Aplicaciones populares como Dropbox, Foursquare, HootSuite utilizan en su infraestructura servicios de AWS. De las ofertas internacionales, es una de las más importantes de computación en la nube considerado como un pionero en este campo que compite directamente con servicios como Microsoft Azure [8] y Google Cloud Platform [9]. AWS ofrece herramientas en diferentes categorías: base de datos, contenedores, robótica y machine learning, entre otras. En este trabajo se utiliza el servicio de cloud computing Amazon EC2 (Elastic Compute Cloud) de la capa gratuita de AWS que

proporciona capacidad cómputo en la nube y todo lo necesario para la creación de instancias, el mantenimiento y el escalamiento de las mismas.

4. Benchmark

Se propuso generar un benchmark propio analizando distintos parámetros, para luego poder relacionarlos y representar a los Cloudlets del simulador. Durante el proceso de investigación de la herramienta CloudSim la realización de dos experimentos permitieron encontrar el parámetro adecuado a modificar.

En el primer experimento se intentó utilizar la propiedad *FileSize*. En un principio, se pensó que al variar este parámetro daría como resultado diferentes tiempos de respuesta de finalización de la tarea que realiza el simulador. En las diferentes pruebas se obtuvo el mismo tiempo de respuesta, con lo que se concluyó que la propiedad *FileSize* no es de utilidad para el fin de este trabajo. En el segundo experimento se utilizó la propiedad *Length*. Como resultado, los tiempos de respuesta de los cloudlets varían proporcionalmente con la cantidad de instrucciones que se indican en este parámetro.

En consecuencia se generó un benchmark para conseguir una forma de obtener cantidades de instrucciones de diferentes tareas y así variar la propiedad *Length*, con lo cual se lograrán tiempos de respuesta diferentes y así poder realizar un análisis con respecto a los tiempos de respuesta en AWS. Teniendo en cuenta los experimentos desarrollados anteriormente, se propone generar un programa en lenguaje de programación C que represente la multiplicación de una matriz por otra, es decir:

$$R[n * n] = A[n * n] * B[n * n] \quad (1)$$

Este programa es utilizado como referencia para comparar en AWS y CloudSim. En la siguiente sección, está el detalle de cómo se establece la relación entre el entorno simulado y el ejecutado en la nube de AWS.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <sys/time.h>
4  #include "matrices.h"
5
6  int *A, *Ai, *R;
7  double timetick;
8
9  int main(int argc, const char* argv[]){
10     if (argc < 2){
11         printf("Falta un parametro.\n");
12         printf("1. Dimension: 512, 1024 y 2048\n");
13         return 0;
14     }
15     N = atoi(argv[1]);
16     int i, j, k;
17     double timetick;
18
19     A = declaration_matrix(A, N);
20     Ai = declaration_matrix(Ai, N);

```

```

21     R = declaration_matrix(R, N);
22     init_matrix(A, R, Ai, N);
23     multi_matrix(R, A, Ai, N);
24     print_matrix(R, N, 0);
25     free_matrix(A, R, Ai);
26 }

```

4.1. Construcción del Benchmark

Por lo explicado anteriormente, el benchmark está definido en 5 módulos que deben mantener relación con CloudSim. Cada módulo corresponde a un cloulet y tienen la característica que cada uno de ellos es incremental al anterior. Es decir que los módulos van incorporando funcionalidades, como se puede ver en el **Cuadro 1**, y de esta forma se logra que la propiedad **Length** del cloulet varíe en cada uno de ellos. Los 5 módulos resuelven el problema planteado anteriormente de resolución de dos matrices de $N \times N$.

Cuadro 1.

Módulo	Métodos Invocados
1	Declaración de matriz A
2	Declaración de matriz A y A_i
3	Declaración de matriz A, A_i y R
4	Declaración de matriz A, A_i y R Inicialización de Matriz A, A_i y R
5	Declaración de matriz A, A_i y R Inicialización de Matriz A, A_i y R Multiplicación de matriz A y A_i almacenamiento de resultado en matriz R Liberación de memoria de matrices A, A_i y R .

Utilizando los módulos del **Cuadro 1** se midió únicamente la cantidad de instrucciones de procesamiento que implementan cada uno. Para ello, en primer lugar se trató que el código no posea referencias a operaciones de llamadas al sistema operativo o de entrada/salida. Luego, se obtuvo mediante el Debug de CodeBlocks la cantidad de instrucciones de los módulos. Posteriormente, se midió el crecimiento de los mismos al incrementar el tamaño de las matrices multiplicadas, observando con especial atención las estructuras de repetición **for(int i=0, i_j=N, i++)**; A modo de ejemplo, se explica el módulo 1. El módulo 1 está compuesto por la siguiente funcionalidad:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <sys/time.h>
4  #include "matrices.h"
5  #define N 512
6
7  int *A, *Ai, *R;
8

```

```

9  int main(int argc, const char* argv[]){
10     A = declaration_matrix(A, N);
11 }

```

Para poder contar la cantidad de instrucciones de cada módulo, se utilizó el programa de computadora CodeBlocks, el cuál tiene incorporada las siguientes herramientas para el análisis del código:

1. Disassembly
2. Watches
3. Memory dump

Estas herramientas permiten observar las instrucciones de máquina generadas por el compilador y poder realizar un conteo de las mismas. Utilizando el debugger de CodeBlocks se obtuvo para el módulo 1 lo siguiente:

```

TAMAÑO DEL MAIN: 13 Inst
TAMAÑO DE FUNCIONES: 13 Inst
TAMAÑO TOTAL DE CODIGO: 26 Inst

```

Este proceso se replica para los módulos siguientes teniendo en cuenta el tamaño de la matriz a multiplicar. Es decir, se mide la cantidad de instrucciones de cada módulo para $N = 512$, $N = 1024$ y $N = 2048$. Una vez obtenida la cantidad de instrucciones de cada módulo, en el entorno de CloudSim, se puede traducir como la cantidad de instrucciones de cada cloudlet dado en millones de instrucciones (MIPs). Se modifica la escala con la que se está trabajando, por ende, si se necesita modelar 10 instrucciones en Cloudsim se cargan como si fuesen 10 MIPs.

5. Ejecución de Benchmark y Mediciones

5.1. Mediciones de Tiempo de Respuesta en AWS

Se ejecutan los módulos definidos en el el Cloud AWS y se simulan las tareas que representan a los mismo módulos en CloudSim. En ambos casos se miden los tiempos de respuesta obtenidos a fin de poder analizar los resultados y llegar a conclusiones.

5.2. Configuración del Entorno en CloudSim

Para llevar a cabo la simulación es necesario configurar las propiedades en CloudSim lo más cercano al entorno real de AWS, en donde se realizaron las mediciones de tiempo de repuesta. Para esto, se configuró de la siguiente manera la máquina virtual del simulador:

```

1  // VM description
2  int vmid = 0;
3  int mips = 480; // Deberia configurarse en 4800
4  long size = 10000; // image size (MB)
5  int ram = 512; // vm memory (MB)
6  long bw = 1000;
7  int pesNumber = 1; // number of cpus

```

De la instancia de AWS se obtuvieron algunos datos de importancia para la simulación como se puede ver en la **Figura 2**.

```

ubuntu@ip-172-31-27-75:~$ cat /proc/cpuinfo
processor       : 0
vendor_id      : GenuineIntel
cpu family     : 6
model          : 63
model name     : Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz
stepping       : 2
microcode     : 0x3c
cpu MHz        : 2400.078
cache size     : 30720 KB
physical id    : 0
siblings       : 1
core id        : 0
cpu cores      : 1
apicid         : 0
initial apicid : 0
fpu            : yes
fpu_exception : yes
cpuid level    : 13
wp             : yes
flags          : fpu vme de pse tsc mtr pae mce cx8 apic sep ntrr pxe mca
               : cpld bpl xtprlogy pni pmlmulgt sse3 fma ckl6 pcid sse4_1 sse4_2 x2apic
               : mwaitx single kaiser fsgbase kmml ssw2 smep bmi2 erms imrpuid rtwnopt
               : bugn
               : cpu_mitdown spectre_v1 spectre_v2 spec_store_bypass l1tf
bogomips       : 4800.15

```

Figura 2. Especificaciones técnicas de la máquina virtual en AWS.

CloudSim soporta un máximo de 1000 MIPS. Como se logró determinar que los tiempos de respuesta se calculan con la relación directa entre la cantidad de instrucciones y los MIPS del procesador, se asignó el valor 480 (valor relacionado con los BogoMips) sabiendo que los resultados obtenidos serán 10 veces mayores. Por otro lado, el tiempo con el que se resuelven las tareas en el simulador para el caso de cloudlets limitados por tareas cortas y con un valor muy alto de MIPS, es menor a 1. Este valor coincide con el tiempo entre SimEvents por defecto que establece cuál es el muestreo entre eventos. Debido a esto, se tuvo que reconfigurar este parámetro para obtener mayor resolución en la escala temporal al medir tiempos de tareas muy pequeñas.

5.3. Configuración del Cloudlet

Se modifica el parámetro “length” por la cantidad de instrucciones obtenidas de los cálculos realizados en MIPS como se explicó anteriormente.

```

1 // Cloudlet properties
2 int id = 0;
3 long length = 26;
4 long fileSize = 20;
5 long outputSize = 20;
6 UtilizationModel utilizationModel =
7     new UtilizationModelFull();

```

Se ejecutó la simulación en CloudSim y se obtuvo lo siguiente salida del simulador:

```

===== OUTPUT =====
Cloudlet ID   STATUS   Start Time   Finish Time
0            SUCCESS   0,0001       0,0543

```

Los tiempos de respuesta obtenidos deben ser ajustados para contemplar la limitación con los MIPs del procesador y a su vez, considerar la cantidad de instrucciones como si fueran millones de instrucciones, entonces para el ejemplo mostrado se obtiene lo siguiente:

$$Respuesta[s] = \frac{0,0542}{1,000,000 \times 10} = 0,0000000542 \quad (2)$$

Este proceso se replicó para todos los cloudlets que fueron generados.

5.4. Configuración y Ejecución en Cloud AWS

Se creó una instancia en alguna de las zonas de disponibilidad de AWS y se realizaron 5 ejecuciones de cada módulo y se obtuvieron diferentes tiempos de respuesta. Para las conclusiones se toman los tiempos promedios. Por ejemplo para los módulos con N=512 se obtuvo los siguientes datos que se pueden ver en el *Cuadro 2*.

Cuadro 2. Tiempos de respuesta para matriz de N = 512.

Ejecuciones	Módulo 1	Módulo 2	Módulo 3	Módulo 4	Módulo 5
T1	0,000035	0,000038	0,000041	0,007840	0,664568
T2	0,000033	0,000040	0,000041	0,007368	0,662680
T3	0,000031	0,000036	0,000039	0,007665	0,663200
T4	0,000035	0,000038	0,000038	0,007463	0,669473
T5	0,000029	0,000033	0,000042	0,007523	0,668542
Promedio	0,0000326	0,000037	0,0000402	0,0075718	0,6656926

6. Presentación de Resultados

A continuación, se presentan los resultados obtenidos, de forma tabulada como se puede ver en el *Cuadro 3*.

En las imágenes siguientes se grafica la relación entre los tiempos de ejecución en CloudSim y en AWS para todas las tareas implementadas por cada uno de los módulos, generando curvas distintas según el tamaño de las tareas. Ésto se puede ver en la *Figura 3*.

Luego, debido a que los dos primeros módulos incluyen bajo nivel de procesamiento en comparación a los siguientes, se hizo foco en los posteriores para una mejor apreciación de los resultados como se vé en la *Figura 4*.

Cuadro 3. Comparación de tiempos de respuestas en CloudSim y Clous AWS

Módulos	#Instrucciones	MI	Tiempo de Respuesta en CloudSim	Tiempo de Respuesta en Cloud AWS	Relación m/AWS	CloudSi-
1 - N=512	26	0,000026	0,00000000542	0,0000326		0,0001662
2 - N=512	44	0,000044	0,00000000917	0,000037		0,0002477
3 - N=512	62	0,000062	0,0000001292	0,0000402		0,0003213
4 - N=512	14942351	14,942351	0,0031298979	0,0075718		0,4111294
5 - N=512	6054740197	6054,740197	1,261404208	0,6656926		1,8948749
1 - N=1024	13	0,000013	0,00000000271	0,000036		0,0000752
2 - N=1024	44	0,000044	0,00000000917	0,000038		0,0002412
3 - N=1024	62	0,000062	0,0000001292	0,000065		0,0001987
4 - N=1024	59768975	59,768975	0,01245186979	0,038918		0,3199514
5 - N=1024	48378151141	48378,151141	10,07878149	5,292631		1,9043046
1 - N=2048	13	0,000013	0,00000000271	0,000035		0,0000774
2 - N=2048	44	0,000044	0,00000000917	0,000039		0,000235
3 - N=2048	62	0,000062	0,0000001292	0,000045		0,000287
4 - N=2048	239075471	239,075471	0,04980738979	0,137613		0,3619381
5 - N=2048	386786132212	386786,132212	80,58044421	42,66248		1,8887895

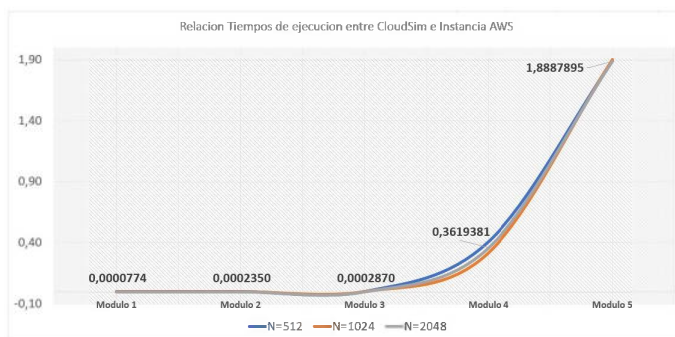


Figura 3. Relación entre el tiempo de ejecución CloudSim y el tiempo de ejecución AWS.

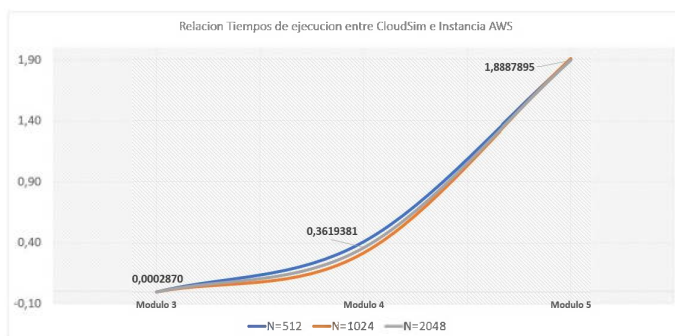


Figura 4. Relación entre el tiempo de ejecución CloudSim y el tiempo de ejecución AWS en tareas largas.

7. Conclusiones

Se desarrollan las conclusiones en base a las pruebas y análisis realizados. A nivel general, se revela la importancia de contar con una herramienta para modelar arquitecturas de Cloud Computing permitiendo parametrizar una vasta gama de componentes y servicios existentes. Luego, en el caso particular de las pruebas realizadas tomando como parámetro los tiempos de ejecución, se comprueba que, la relación entre los tiempos de respuesta en AWS y en el simulador CloudSim sigue un comportamiento lineal. Aunque, al tratarse de un experimento acotado, si se extiende el rango de pruebas incrementando tamaños seguramente la curva tienda a un valor máximo y se mantenga en ese valor. Además, se comprueba que el valor de la relación entre los tiempos de respuesta en el Cloud AWS y en el simulador CloudSim depende de la tarea realizada.

A partir de las dos últimas conclusiones particulares, se deriva que para poder estimar el tiempo de procesamiento de una tarea específica (y por ende su costo asociado), es imperativo poder determinar una muestra mínima de esa tarea (ejemplo: para renderizar una imagen de 200 MPíxeles sería necesario estimar las instrucciones del renderizado de una imagen de 1 MPíxel). Luego simular la ejecución en CloudSim, realizar esa tarea en un Cloud real (ej: AWS) para determinar la relación de tiempos. Es decir, realizar una calibración del simulador generado y de esta manera se podrían predecir resultados para tareas de mayores tamaños.

Referencias

1. Buyya, R., Broberg, J., Gościński, A. (Eds.). Cloud computing: principles and paradigms. Hoboken, N.J: Wiley. (2011)
2. Alberto Crespo Guzmán. Um Estudo Da Simulação de Serviços da Amazon EC2 Utilizando o Simulador CloudSim. (2011)
3. Amazon Web Services, <https://aws.amazon.com/es/what-is-aws/>
4. Melbourne Clouds Labs, <http://www.cloudbus.org/cloudsim/>
5. Patiño, Paláez, Granda, Segovia. Diseño y Simulación de un Data Center Cloud Computing que cumpla con la norma PCI-DSS. (2019)
6. Rosatto, D.; Bond, R.; Encinas, D.; Morales, M. "Modelado y simulación de arquitecturas de cloud computing con cloudsim: comunicación entre entidades". XXIII Congreso Argentino de Ciencias de la Computación (CACIC 2017). La Plata, Argentina.
7. Calheiros, R., Ranjan, R., Buyya, R. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Journal Software-Practice Experience. Volume 41, John Wiley Sons. (2011)
8. Microsoft Azure, <https://azure.microsoft.com/es-es/overview/>
9. Google Cloud Platform, <https://cloud.google.com/>