PROYECTO DE INVESTIGACIÓN

Semana 15 - Container Cloudsim y CloudSim Plus

Estudiante:

Alejandro Naranjo

Curso:

BIT-28 Sistemas Operativos II

Profesor:

Carlos Andres Mendez Rodriguez

Noviembre, 2024

Sede San Pedro

Tabla de contenidos

Índice de Figuras

Índice de Tablas

1. Definición del tema

CloudSim ayuda a estudiar la computación en la nube sin costosas configuraciones de la vida real. Esta investigación analiza cómo utilizar mejor CloudSim para garantizar que los recursos de la nube se utilicen de manera eficiente, manteniendo el sistema siempre disponible, capaz de crecer según sea necesario, de manera rápida y segura. Se evaluará la influencia de varios métodos de distribución de carga en la eficiencia del sistema, recomendando mejoras en función de los resultados obtenidos.

2. Objetivo general

Diseñar y evaluar una arquitectura de simulación en CloudSim que integre contenedores, con el propósito de optimizar la administración de recursos en términos de disponibilidad, rendimiento, escalabilidad y seguridad.

3. Objetivos específicos

1. Analizar los aspectos fundamentales de alta disponibilidad, alto rendimiento, escalabilidad y seguridad en la administración de contenedores en CloudSim.
2. Diseñar una arquitectura robusta en CloudSim que incorpore los principios mencionados para optimizar el uso de recursos.
3. Evaluar el impacto de diferentes algoritmos de balanceo de carga en el rendimiento del sistema simulado.
4. Determinar los criterios de evaluación y métricas clave de rendimiento, como:
    - Tiempo de respuesta.
    - Distribución del tráfico.
    - Tiempo de inactividad.

5. Proponer mejoras en las estrategias de simulación y administración de recursos basadas en los resultados obtenidos durante las simulaciones.

4. Plan de Trabajo

4.1 Fase 1: Investigación y Preparación

I. **Actividades:**

A. Revisión de información sobre CloudSim y su extensión Container CloudSim

B. Análisis algoritmos de balanceo de carga (ej, Least Loaded, Round Robin, Bee Load).

C. Identificación de métricas clave para el rendimiento (ej, tiempo de respuesta, distribución del tráfico, tiempo de inactividad).

II. **Entregables:**

A. Documentación inicial

B. Lista de métricas clave y su relevancia.

C. Elección de algoritmo de balanceo de carga.

4.2 Fase 2: Configuración del Entorno de Simulación

I. **Actividades:**

A. Instalación y configuración del entorno de desarrollo (Eclipse, IntelliJ o Microsoft Visual Studio).

B. Configuración de Container CloudSim: Importación de bibliotecas y configuración inicial.

C. Creación de un escenario básico con hosts, VMs y contenedores.

II. **Entregables**:

A. Código funcional de simulación básica.

B. Entorno de desarrollo configurado correctamente.

4.3 Fase 3: Implementación de Balanceo de Carga

I. **Actividades**:

A. Diseño de algoritmos para asignación de recursos.

B. Incorporación del algoritmo al código.

C. Validación de asignaciones correctas a través de logs.

II. Entregables:

A. Clase personalizada de balanceo y asignación de recursos.

B. Simulación funcional.

4.4 Fase 4: Ejecución y Análisis de Resultados

I. **Actividades**:

A. Ejecución de simulaciones con diferentes cargas.

B. Recolección de métricas clave de rendimiento.

C. Comparación de resultados con y sin balanceo de carga.

II. **Entregables**:

A. Reporte de métricas.

B. Gráficos y visualizaciones (Se intentará mostrar en gráficos).

4.5 Fase 5: Documentación y Presentación

I. **Actividades**:

A. Finalización de la documentación con los resultados finales.

B. Preparación de la presentación con resultados y recomendaciones.

II. **Entregables**:

A. Informe completo del proyecto.

B. Presentación.

5. Metodología de investigación

5.1 Tipo de investigación

I. Investigación experimental: Se realizarán simulaciones de un entorno controlado para poder analizar el comportamiento de diferentes métricas a evaluar de Container CloudSim.

5.2 Enfoque

I. **Cuantitativo:** Se medirán métricas clave como el tiempo de respuesta, distribución de tráfico, y tiempo de inactividad para evaluar la eficiencia en la simulación.

5.3 Procedimientos

I. **Desarrollo iterativo:** Cada fase del proyecto se completa antes de avanzar al siguiente, de esta forma se permite realizar ajustes en función de los resultados.

5.4 Criterios de Evaluación

I. Rendimiento del sistema simulado.

II. Adecuación de las herramientas utilizadas.

III. Análisis y comparación de resultados.

6. Herramientas y recursos necesarios

6.1 Herramientas de Software

I. **Container CloudSim:** Para la simulación y evaluación del rendimiento en entornos de contenedores.

II. **Eclipse IDE, IntelliJ IDEA o Microsoft Visual Studio:** Entorno de desarrollo integrado para escribir y ejecutar código Java.

III. **Java Development Kit (JDK):** Versión 8 o superior.

IV. **Bibliotecas necesarias:**

   A. cloudsim-3.0.3.jar
   B. containercloudsim.jar
   C. commons-math3.jar

V. **JFreeChart (Experimental):** Para visualización de métricas en gráficos.

6.2 Recursos Humanos

I. **Estudiante familiarizado con:**

   A. Programación en Java.
   B. Fundamentos de simulación en la nube.

6.3 Recursos Documentales

I. Artículos y manuales sobre Container CloudSim.

II. Documentación de algoritmos de balanceo de carga.

6.4 Recursos Hardware

I. **Computadora con al menos:**

A. Procesador: Intel I5 o superior (Equivalente en AMD).

B. RAM: 8 GB.

C. Espacio en disco: 500 MB para bibliotecas y datos.

7. Marco Teórico

7.1 Computación en la nube

7.1.1 Definición

La computación en la nube se determina por su capacidad de brindar servicios con mucha flexibilidad y escalabilidad. Según Singh et al. (2020), *"Cloud computing is a heterogeneous architecture, on-demand self-service, broad network access and multiple client devices, resource pooling, rapid elasticity, and measured service with the pay-per-use business model"* (p. 730). Estas características no únicamente facilitan la adaptación a distintas necesidades de los usuarios, sino que también fomentan la optimización de recursos tecnológicos.

En este entorno, los autores resaltan la relevancia de políticas de consolidación de máquinas virtuales y técnicas como DVFS para reducir el consumo de energía en los centros de datos, lo que resulta prioritario para mejorar la sostenibilidad de los distintos servicios. Esto subraya la importancia de combinar soluciones eficientes en infraestructuras en la nube, especialmente en entornos con alta demanda computacional.

7.2 CloudSim

7.2.1 Definición

CloudSim es un marco de simulación generalizado y extensible creado para poder modelar, simular y experimentar con infraestructuras y servicios de computación en la nube. Su primer y más importante objetivo es facilitar a investigadores y desarrolladores centrarse en situaciones específicas de creación de sistemas sin tener que preocuparse por los detalles de bajo nivel contraídos con las infraestructuras y servicios basados en la nube. Esta herramienta

brinda la evaluación de hipótesis y el ajuste de rendimiento en un entorno controlado y reproducible, lo que resulta útil tanto para clientes como para proveedores de servicios en la nube.

Como se menciona, *"en el caso de la computación en la nube, donde el acceso a la infraestructura implica costos reales, los enfoques basados en simulación ofrecen beneficios significativos, permitiendo a los clientes probar sus servicios en un entorno repetible y sin costos, además de ajustar cuellos de botella antes del despliegue en nubes reales"*. (CLOUDS Lab, 2009).

7.3 CloudSim Plus

7.3.1 Definición

Cloudsim plus es *"an open source simulation framework that pursues conformance to software engineering principles and object-oriented design in order to provide an extensible, modular and accurate tool."* (Silva Filho et al., 24, 1) Que está basado en su padre CloudSim el cual fué comentado anteriormente.

La idea principal era que en su versión CloudSim 3, habían muchos errores, bugs, problemas de optimización que luego los creadores de CloudSim Plus mejoraron, e hicieron un entorno más amigable para todos mejorando dichos problemas pero más bien creando mayores posibilidades que el original.

7.3.2 Funciones

Existen muchos tipos de funciones que puede realizar CloudSim Plus pero de sus principales funciones y características se encuentran:

- Mucho más fácil de usar que su padre CloudSim, ya que está planeado y planteado también para personas con 0 conocimiento de CloudSim con ejemplos más básicos y demás.
- Simulaciones en conjunto que toman en cuenta la potencia y red.
- Escalado de máquinas virtuales tanto vertical como horizontal.
- Cálculos de consumo de energía, utilización de CPU, ancho de banda, Ram y muchas otras más.

7.4 Container CloudSim

7.4.1 Definición

Los contenedores, como **Docker**, son entornos ligeros que permiten ejecutar aplicaciones de manera separada. Distintos de las máquinas virtuales (VMs), los contenedores trabajan sobre el mismo sistema operativo subyacente, lo que los hace más trabajadores en términos de recursos, ya que no requieren un sistema completo para cada instancia. Esto permite una mayor cantidad y un arranque más rápido en comparación con las VMs, lo cual es fundamental para la escalabilidad en entornos de nube.

Container CloudSim permite realizar simulaciones sobre estos contenedores, lo que ofrece una implementación más precisa y representativa de cómo se administran y escalan los recursos en los contenedores modernos.

7.4.2 Ventajas

- **Uso eficiente de recursos**: Los contenedores son menos pesados y más rápidos de inicializar que las máquinas virtuales. Esto resuena en un uso más cómodo de los recursos y en menores costos operativos.
- **Escalabilidad**: Los contenedores pueden aumentar rápidamente, lo que es ideal para necesidades donde se requiere ajustar la capacidad según la demanda de recursos.

- **Aislamiento**: Aunque los contenedores se componen del mismo sistema operativo, aún se mantienen aislados unos de otros, lo que les facilita ejecutar aplicaciones de manera independiente sin interrupciones.

7.4.3 Principales Componentes

- **Contenedores**: En lugar de usar máquinas virtuales para correr las aplicaciones, **Container CloudSim** simula la creación y gestión de contenedores, lo que cede el paso a modelar el rendimiento y la colocación de los recursos en un sistema regido por contenedores.
- **Hosts**: Los hosts en Container CloudSim son los recursos físicos (servidores) que pueden contraer tanto contenedores como máquinas virtuales.
- **Servicios de infraestructura**: Container CloudSim expande la infraestructura de CloudSim para brindar servicios específicos de contenedores, como la creación y manejo de contenedores y la distribución dinámica de recursos.

7.5 Balanceo de Cargas

7.5.1 Definición

El balanceo de carga es un proceso mediante el cual el tráfico saliente de una red se distribuye de manera eficiente a través de múltiples enlaces disponibles. (Teixeira, n.d.) Este mecanismo tiene como principal función optimizar el uso de los recursos de red o muchos otros factores para lograr aumentar la capacidad total de transmisión y garantizar una mayor disponibilidad de los servicios en el entorno que se estén trabajando los datos.

7.5.2 Tipos

"*Round Robin algorithm is one of the oldest, simplest, fairest and most widely used scheduling algorithms, designed specifically for time-sharing systems. In this algorithm, the processes share the CPU time by allocating a slice of time to each process*" (Institute of Electrical and Electronics Engineers, 2019, 1-7). Esto nos da a entender que es un tipo de balanceo de carga el cuál nos permite distribuir el tráfico que queremos de manera cíclica dentro de un gran número de servidores.

Esto nos ayuda para poder mantener un flujo constante y que no se sobrecarguen nuestros servidores y que se reparta todo equitativamente para evitar problemas a futuro cuando tengamos que sacar a producción nuestra información y no existan problemas.

Least Loaded hace alusión a enviar el flujo de tráfico por servidores que están menos cargados, esto hace referencia al evitar sobrecargar servidores demás para así poder tener un balanceo de carga más tranquilo y que este nos brinde facilidad en tiempos de respuesta y demás movimiento de información para que no se sobrecargue y se caiga la red.

Ant Colony Optimization utiliza el comportamiento de las hormigas ya que si bien vemos en perspectiva aérea el comportamiento de las rutas de las hormigas, siempre intentan seguir el camino menos largo y complejo, pues este balanceo de carga realiza las mismas acciones, buscando cual es el servidor en el que mejor señal, datos espacio, y todo lo posible se encuentre en mejor estado para específicamente los datos que se están enviando.

7.6 Métricas de Medición

7.6.1 Definición

Las métricas de medición son clave porque "*conocer la capacidad de rendimiento aporta información para orientar el proceso de planificación y control*" (Diez-Silva et al., 2012, 2). Dando esto a entender que si tenemos métricas reales de medición de Datos o cualquier tipo que se requiera nos puede llevar hacia un mejor flujo y manejo para poder llegar al objetivo deseado de la mejor manera.

Con esto en mente se puede tomar las mejores rutas de acción para los datos para lograr llegar a los resultados esperados ya que con una mala implementación de algoritmos y flujos de datos podrían llegar a ser efectivos más no los resultados esperados reales ya sea por milésimas de segundo de diferencia que en un ambiente estricto serían determinantes para la validez de la información suministrada.

8. Análisis de resultados

8.1 Container Cloudsim

La simulación puede ser vista en:

https://github.com/AleNaranjo-Git/cloudsim.git

En la ruta: modules > cloudsim-examples > src > main > java > org > cloudbus > cloudsim > examples > container > ContainerCloudSimExample1.java

La simulación de Container cloudsim se enfoca en una arquitectura que dentro de sí poseé Cloudlets, Host, VMs y Contenedores para estar sujeta a una buena eficiencia de asignación de recursos en la computación en la nube.

En dichas simulaciones los Cloudlets representan las tareas que van a ser asignadas con sus diferentes cargas, los Host son los servidores que se utilizarán, las VMs funcionan en esta simulación como el medio por el cual se pueden alojar los distintos contenedores y los contenedores son los que ejecutan los cloudlets.

La primer pincelada acerca de la optimización de recursos escalabilidad y demás temas que se pueden tocar son las políticas de asignación que existen dentro de la simulación, tenemos dos tipos, el primero es Allocation Policy Simpler dicha política de asignación de VMs es la base simple de cualquier simulación en CloudSim no es nada muy complejo y realmente sus resultados son bastante estándares de lo que se esperaría ya que no cumplen una función muy avanzada.

Después se utiliza Power Container Vm Allocation Policy Migration Abstract Host Selection, dicha política lo que realiza es que dependendiendo de que tan sobreutilizado o poco utilizado

esté un host, es como se van a ir colocando los diferentes contenedores de manera dinámica, dicha locación ayuda mucho y mejora la consolidación y optimización del consumo de los recursos.

Figura 1

Resultados simulación Container CloudSim

```
========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID    Time    Start Time    Finish Time
    1        SUCCESS        3             1        0,05        0,02        0,07
    2        SUCCESS        3             1        0,05        0,02        0,07
    3        SUCCESS        3             1        0,05        0,02        0,07
    4        SUCCESS        3             1        0,05        0,02        0,07
    5        SUCCESS        3             1        0,05        0,02        0,07
    10        SUCCESS        3             2        0,05        0,02        0,07
    12        SUCCESS        3             2        0,05        0,02        0,07
    14        SUCCESS        3             2        0,05        0,02        0,07
    15        SUCCESS        3             2        0,05        0,02        0,07
    17        SUCCESS        3             3        0,05        0,02        0,07
    22        SUCCESS        3             4        0,05        0,02        0,07
    26        SUCCESS        3             5        0,05        0,02        0,07
    27        SUCCESS        3             5        0,05        0,02        0,07
    30        SUCCESS        3             6        0,05        0,02        0,07
    31        SUCCESS        3             6        0,05        0,02        0,07
    32        SUCCESS        3             6        0,05        0,02        0,07
    33        SUCCESS        3             7        0,05        0,02        0,07
    34        SUCCESS        3             7        0,05        0,02        0,07
    37        SUCCESS        3             8        0,05        0,02        0,07
    38        SUCCESS        3             8        0,05        0,02        0,07
    41        SUCCESS        3             9        0,05        0,02        0,07
    46        SUCCESS        3             10        0,05        0,02        0,07
    49        SUCCESS        3             11        0,05        0,02        0,07
    50        SUCCESS        3             11        0,05        0,02        0,07
    6        SUCCESS        3             1        16,43        0,02        16,45
    8        SUCCESS        3             1        16,43        0,02        16,45
    9        SUCCESS        3             2        16,43        0,02        16,45
    11        SUCCESS        3             2        16,43        0,02        16,45
    13        SUCCESS        3             2        16,43        0,02        16,45
    16        SUCCESS        3             2        16,43        0,02        16,45
    20        SUCCESS        3             3        16,43        0,02        16,45
    21        SUCCESS        3             4        16,43        0,02        16,45
    23        SUCCESS        3             4        16,43        0,02        16,45
    24        SUCCESS        3             4        16,43        0,02        16,45
    39        SUCCESS        3             8        16,43        0,02        16,45
    25        SUCCESS        3             5        16,43        0,02        16,45
    25        SUCCESS        3             5        16,43        0,02        16,45
    25        SUCCESS        3             5        16,43        0,02        16,45
    28        SUCCESS        3             5        16,43        0,02        16,45
    29        SUCCESS        3             6        16,43        0,02        16,45
    44        SUCCESS        3             10        16,43        0,02        16,45
    47        SUCCESS        3             10        16,43        0,02        16,45
    48        SUCCESS        3             11        16,43        0,02        16,45
    7        SUCCESS        3             1        32,82        0,02        32,84
    19        SUCCESS        3             3        32,82        0,02        32,84
    36        SUCCESS        3             8        32,82        0,02        32,84
    40        SUCCESS        3             9        32,82        0,02        32,84
    42        SUCCESS        3             9        32,82        0,02        32,84
    43        SUCCESS        3             9        32,82        0,02        32,84
    45        SUCCESS        3             10        32,82        0,02        32,84
    35        SUCCESS        3             7        32,82        0,02        32,84
```

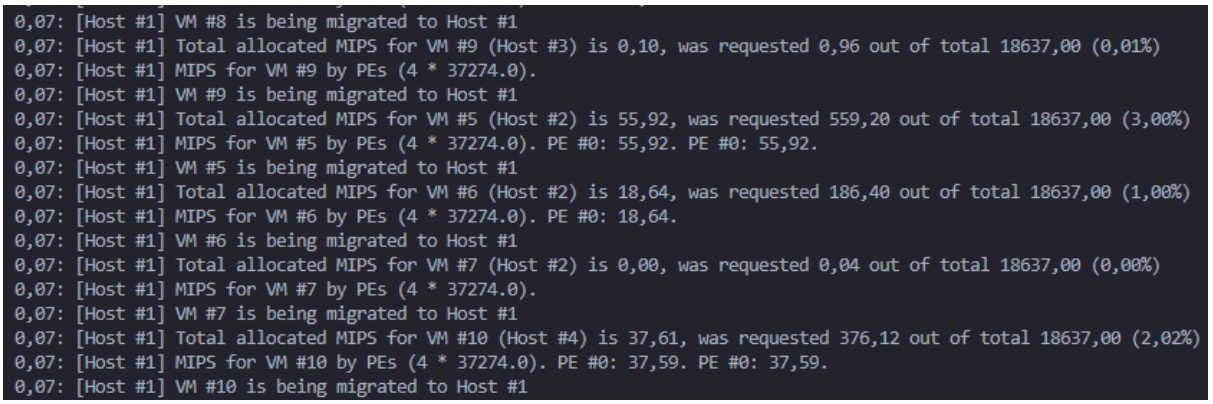
Nota. Datos de salida de la simulación.Elaboración propia

En esta simulación lo que se utilizó para el balanceo de cargas fueron 2 distintas para distintos elementos de la primera fué Selection Policy Maximum Usage para las máquinas virtuales. Esta política lo que nos permite realizar es saber cual es la mejor opción de máquina virtual a la hora de realizar una migración cuando un host está sobrecargado. Dicha política lo que nos permite es aliviar los recursos eficazmente cuando un host se sobrecarga.

Como bien se puede ver en la Figura 2 podemos ver que las VMs están realizando migraciones. Con esto podemos ver una técnica usual el cual nos permite equilibrar nuestra carga de trabajo y de esta forma, se optimizan los recursos los cuales nos ayudan a realizar mejores simulaciones de computación en la nube.

Figura 2

VMs migrando a otro Host



Nota. VMs migrando a otros Host. Elaboración propia.

Gracias a estas salidas de datos podemos ver que si se están aplicando las políticas de migración de utilizar el Maximum Usage para que de esta forma no se sobrecarguen algunos Hosts.

Después tenemos la siguiente política que sería Selection Policy FirstFit está política es bastante simple, se utiliza para poder acomodar los Cloudlets a los host en el primero que logra complir sus capacidades de recursos necesarios. Por dar un ejemplo si se tiene un bulto de 100 kg y meto una piedra de 10 kg si cabe entonces seleccionó ese bulto. Cabe recalcar que esta es una política que puede generar problemas en simulaciones de mayor escala.

Porque si bien nuestra simulación no llega a esos márgenes de error que pasa si después tengo una piedra que pesa 100kg, ya no habría un bulto al cual meterlo, y esto también aplica para nuestra simulación, si un cloudlet requiere de mas potencia pero ya no existe un host capaz de sostener dicha carga. vamos a tener problemas en nuestras simulaciones por lo tanto dicha política es buena en cargas pequeñas pero si ya se requieren de más potencia habría que intentar utilizar otro tipo de política de balanceo.

Para analizar bien nuestras métricas de evaluación de dichas políticas para balanceo de cargas podemos notar el siguiente patrón, si bien dichos balanceos nos ayudan a evitar la sobrecarga y así que se nos rompa la simulación, bien podemos ver en la **Figura 1** los tiempos de ejecución nos muestran que hay algunas tareas que realmente son muy ligeras entonces pueden cargar muy rápido pero hay otras que más bien duran mucho entonces significa que están muy cargadas.

Esto lo que nos enseña es que realmente las políticas utilizadas intentan utilizar la menor cantidad de energía para el mejor equilibrio posible, ya que como bien podemos ver después de unas simulaciones algunas VMs están muy sobrecargadas y esto nos lleva a tiempos de ejecución cada vez más altos todavía. Con esto en mente, el desarrollo se podría seguir

aumentando para poder llegar después a políticas que bajen todavía más los tiempos de carga y se mantenga todo más estable.

8.2 CloudSim Plus

La simulación puede ser vista en:

https://github.com/AleNaranjo-Git/cloudsimplus-examples.git

En la ruta: Source Packages > org.cloudsimplus.examples > VmAllocationPolicyRoundRobinExample.java

La simulación de CloudSim Plus internamente se construye de Hosts, Máquinas Virtuales y Cloudlets. En dicha simulación, se realizó un balanceo de carga distinto que en Container Cloudsim para de esta manera ver distintos tipos de Balanceos, en este caso se utilizó RoundRobin. Para recordar cómo era dicho balanceo es acomodar las tareas (Cloudlets) de manera equitativa entre todos los hosts (Servidores) disponibles en la simulación.

Las constantes utilizadas en dicha simulación se pueden ver en la **Figura 3.** Dichas constantes fueron utilizadas para realizar una simulación básica de cómo funcionará Round Robin y poder demostrarlo de manera más sencilla cuáles serían sus resultados para de esta forma no llegar a los extremos de la complejidad y así no poder demostrar de manera más sencilla.

Figura 3

Constantes utilizadas para la simulación.

```
private static final int HOSTS = 4;
private static final int HOST_PES = 8;

private static final int VMS = 8;
private static final int VM_PES = 2;

private static final int CLOUDLETS = 8;
private static final int CLOUDLET_PES = 2;
private static final int CLOUDLET_LENGTH = 10000;
```

Nota. Constantes. Elaboración Propia.

Gracias a dicho tipo de balanceo podemos ver que a la hora de ejecutar la simulación como se muestra en la **Figura 4** podemos ver que hubo un orden en el que las VMs fueron asignadas a los host de manera equitativa, ya que podemos ver que los host recibieron 2 Vms.

Figura 4

Output Simulación Balanceo Round Robin

```
INFO  0,00: HostSimple: Vm 0 is booting up right away in Host 0/#DC 1, since no startup delay (boot time) was set.
INFO  0,00: VmAllocationPolicyRoundRobin: Vm 0 has been allocated to Host 0/#DC 1
INFO  0,00: Host 1/#DC 1 is powered on.
INFO  0,00: HostSimple: Vm 1 is booting up right away in Host 1/#DC 1, since no startup delay (boot time) was set.
INFO  0,00: VmAllocationPolicyRoundRobin: Vm 1 has been allocated to Host 1/#DC 1
INFO  0,00: Host 2/#DC 1 is powered on.
INFO  0,00: HostSimple: Vm 2 is booting up right away in Host 2/#DC 1, since no startup delay (boot time) was set.
INFO  0,00: VmAllocationPolicyRoundRobin: Vm 2 has been allocated to Host 2/#DC 1
INFO  0,00: Host 3/#DC 1 is powered on.
INFO  0,00: HostSimple: Vm 3 is booting up right away in Host 3/#DC 1, since no startup delay (boot time) was set.
INFO  0,00: VmAllocationPolicyRoundRobin: Vm 3 has been allocated to Host 3/#DC 1
INFO  0,00: HostSimple: Vm 4 is booting up right away in Host 0/#DC 1, since no startup delay (boot time) was set.
INFO  0,00: VmAllocationPolicyRoundRobin: Vm 4 has been allocated to Host 0/#DC 1
INFO  0,00: HostSimple: Vm 5 is booting up right away in Host 1/#DC 1, since no startup delay (boot time) was set.
INFO  0,00: VmAllocationPolicyRoundRobin: Vm 5 has been allocated to Host 1/#DC 1
INFO  0,00: HostSimple: Vm 6 is booting up right away in Host 2/#DC 1, since no startup delay (boot time) was set.
INFO  0,00: VmAllocationPolicyRoundRobin: Vm 6 has been allocated to Host 2/#DC 1
INFO  0,00: HostSimple: Vm 7 is booting up right away in Host 3/#DC 1, since no startup delay (boot time) was set.
INFO  0,00: VmAllocationPolicyRoundRobin: Vm 7 has been allocated to Host 3/#DC 1
```

Nota. Se muestra la salida de la simulación. Elaboración Propia.

Para una visualización más sencilla del proceso que realizó el balanceo se simplificará la salida de datos que tuvo la simulación para poder demostrar su trabajo realizado. Dichos datos se mostrarán en la **Tabla 1** como se puede ver a continuación.

Tabla 1

Salida de datos de la simulación simplificados visualmente.

Host	Asignado
0	Vm 0, Vm 4
1	Vm 1, Vm 5
2	Vm 2, Vm 6
3	Vm 3, Vm 7

Nota. Salida simplificada de datos. Elaboración Propia.

Después, podemos notar que cada cloudlet son tareas que realmente tiene un tamaño de 10,000 MI y luego cada VM tiene 2 PE (Estos son Processing Elements) esto hace que se determine la velocidad que procesan los los cloudlets. Cabe recalcar que podemos ver que cada tarea comienza en 0.1 y terminan en 20.1 segundos.

La fórmula para poder calcular cuánto tiempo se durará en la ejecución es la siguiente.

$$Tiempo\ de\ ejecución = \frac{Longitud\ de\ Tarea}{VM\ PE \times MIPS}$$

Ya con esto podemos ver en la **Figura 5** que al ser una simulación pequeña logramos tener resultados completamente equitativos para de esta forma poder demostrar el funcionamiento correcto de nuestro balanceo de Round Robin.

Figura 5

Salida de información simulación.

SIMULATION RESULTS

Cloudlet ID	Status	DC ID	Host ID	Host PEs CPU cores	VM ID	VM PEs CPU cores	CloudletLen MI	FinishedLen MI	CloudletPEs CPU cores	StartTime Seconds	FinishTime Seconds	ExecTime Seconds
0	SUCCESS	1	0	8	0	2	10000	10000	2	0,1	20,1	20,0
1	SUCCESS	1	1	8	1	2	10000	10000	2	0,1	20,1	20,0
2	SUCCESS	1	2	8	2	2	10000	10000	2	0,1	20,1	20,0
3	SUCCESS	1	3	8	3	2	10000	10000	2	0,1	20,1	20,0
4	SUCCESS	1	0	8	4	2	10000	10000	2	0,1	20,1	20,0
5	SUCCESS	1	1	8	5	2	10000	10000	2	0,1	20,1	20,0
6	SUCCESS	1	2	8	6	2	10000	10000	2	0,1	20,1	20,0
7	SUCCESS	1	3	8	7	2	10000	10000	2	0,1	20,1	20,0

Nota. Salida de información. Elaboración Propia.

Gracias a la simulación, pudimos darnos cuenta de varios aspectos importantes, el primero siendo que los cloudtest tuvieron una eficiencia ya que ninguno tuvo ningún tipo de retraso o tiempo muerto el cuál afectará la demostración del balanceo de carga. Después, se vió evidenciado cómo funciona la política de round robin siendo esta la repartición equitativa de los vm a todos los host.

Si quisiéramos realizar una simulación mucho más robusta para poder demostrar una fehaciente de su rendimiento como balanceo de carga, sería recomendable agregar más métricas de medición, introducir tiempos de arranque o retrasos en la red y variar el número de PEs en las VMs y Hosts para analizar su impacto en el rendimiento.

8.3 Comparación de resultados

A Pesar de que la simulación de container cloudsim es muchísimo más avanzada que la simulación básica de cloudsim plus para demostrar los balanceos de carga, podemos darnos cuenta de varias diferencias entre ellas viendo bien los puntos clave de cada una de sus funciones y de cómo llegan a trabajar más a fondo para su análisis.

Como primera gran diferencia podemos ver que Container Cloudsim utiliza dos políticas distintas para la repartición de VM a los Host, la primera siendo Allocation Policy Simpler que básicamente nos permite tener resultados estándares ya que no es nada muy avanzado. Pero su segunda que es Power Container VM Allocation Policy Migration Abstract Host Selection lo que realiza es optimizar recursos gracias a que migra dinámicamente VMs según que tan utilizados están los Host.

Esto nos lleva a ver que en primera instancia ya la simulación de Cloudsim Plus no las utiliza entonces podría esta llegar a darnos más problemas entre más grande queramos hacer la simulación por eso entre más complejas sean ocuparemos de más políticas para de esta forma regular los parámetros y que no se salgan de control las simulaciones con datos que no son realmente los que queremos o más cercanos llegar a la hora de analizar sus resultados.

Después tenemos estas dos políticas de balanceo Selection Policy Maximum Usage y Selection Policy FirstFit que tienen un aspecto negativo ayq ue la segunda política a pesar de que sirve para entornos de simulación más grandes realmente depende de qué tan crítica sea la tarea vemos en la simulación que no tiene un tiempo de ejecución muy rápido por eso para poder llegar a tener tiempos más bajos en tareas muy pesadas habría que lograr ajustar mejor las políticas para así llegar a tiempos más bajos.

En cambio, CloudSim Plus solo tiene Round Robin que como su principal aspecto negativo es que utiliza todos los host disponibles puedo este no ser necesario entonces lo que realiza esto es un mucho mayor consumo energético del que se debería realmente utilizar ya que este si prende todos los host por más que realmente no sea necesario y uno solo pueda manejar todas las tareas.

Entonces para dar unas diferencias más concisas de sí mismo y dar a conocer que es lo que cada uno da como sus principales puntos son qué Container CloudSim a pesar de que realmente la simulación si es mucho más compleja podemos darnos cuenta de estos aspectos que son políticas mucho más avanzadas y nos permiten trabajar con entornos complejos y dinámicos, algo que round robin no puede. Se centra también en migraciones y consolidación de recursos para así evitar consumir mucha energía y evitar usar Host que están sobrecargados.

Esto está siendo evidenciado con el tipo de balanceo de carga que es el propio round robin. Y como última gran diferencia es que estas políticas nos permiten ser flexibles con cargas que sean cambiantes durante su ejecución y optimizar tanto las tareas que son asignadas como los recursos que se están utilizando para así no malgastar. Luego tenemos por otra parte los aspectos de CloudSim Plus que son políticas mucho más simples y lo que buscan es equilibrio. No considerar factores variables como lo son datos dinámicos con cargas variables o ver temas de consolidación energética.

Y por último es mucho más adecuado para simulaciones pequeñas por lo mismo de que esta política tiende mucho a utilizar host de manera innecesaria entonces entre más grande sea la simulación más recursos estaría utilizando que realmente no se requieren.

Viéndolo ya desde un espectro más amplio, Container CloudSim gracias a sus políticas es mucho más escalable y adecuado para simulaciones más complejas con muchas tareas a

realizar o computación dinámica. Pero como aspecto negativo es que gracias a su cantidad de políticas, su configuración y análisis puede llegar a ser también muy elevado entonces no permitiría su fácil manipulación.

Para CloudSim Plus es menos compleja y mucho más fácil de utilizar pero su aspecto negativo es que entre más grande y compleja se quiera hacer la simulación habría problemas por su falta de políticas.

En conclusión, todas las políticas nos ayudan a demostrar escenarios distintos, pero como bien se pudo evidenciar en esta investigación, todas las políticas tiene sus aspectos negativos como el gasto de recursos innecesarios siendo este el mayor tema principal o una optimización no adecuada para grandes volúmenes de tareas lo cual puede llegar a afectar la validez de los resultados esperados en las simulaciones realizadas.

References

CLOUDS Lab. (2009). *CloudSim: A Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services*. CLOUDS Lab. Retrieved December 1, 2024, from http://cloudbus.org/cloudsim/

D, J.-M., Butt, A. S., Onyema, M. E., & et al. (2021, July 17). Artificial intelligence-based Kubernetes container for scheduling nodes of energy composition. *International Journal of System Assurance Engineering and Management.* https://doi.org/10.1007/s13198-021-01195-8

Diez-Silva, M. H., Pérez-Ezcurdia, A. M., Ramos, G. F. N., & Montes-Guerra, M. I. (2012, Octubre 25). Medición del desempeño y éxito en la dirección de proyectos. *Revista EAN*, *73*, 60-79. 0120-8160

Gholipour, N. N., Ariayan, E. E., & Buyya, R. R. (2020). *A novel energy-aware resource management technique using joint VM and container consolidation approach for green computing in cloud data centers* (Vol. 104). Simulation Modelling Practice and Theory. https://doi.org/10.1016/j.simpat.2020.102127

Hussain, A., Aleem, M., Iqbal, A. M., & Islam, A. M. (2019, August 1). Investigation of Cloud Scheduling Algorithms for Resource Utilization Using Cloudsim. *Computing & Informatics.*, *38*(3), 525-554. https://doi.org/10.31577/cai_2019_3_525

Institute of Electrical and Electronics Engineers. (2019). *2nd International Conference on Computer Applications & Information Security: (ICCAIS' 2019) : 01-03 May, 2019, Riyadh, Kingdom of Saudi Arabia*. IEEE. 10.1109/CAIS.2019.8769534

Institute of Electrical and Electronics Engineers (IEEE), IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology, Velde, V., & Rama, B. (2017). *RTEICT-2017: 2nd IEEE International Conference on Recent*

Trends in Electronics, Information & Communication Technology : 19-20 May 2017 : Proceedings. IEEE. 10.1109/RTEICT.2017.8256824

Khatri, S. K. (Ed.). (2018). *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions): August 29-31, 2018, Venue, Amity University Uttar Pradesh, Noida, India*. IEEE. 10.1109/ICRITO.2018.8748514

Saleh, N., & Mashaly, M. (12). *2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS)* (March ed., Vol. 2020). IEEE. 10.1109/ICICIS46948.2019.9014697

Santra, S., & Mali, K. (2015). *Computer, Communication and Control (IC4), 2015 International Conference on*. IEEE. 10.1109/IC4.2015.7375671

Silva Filho, M. C., Oliveira, R. L., Monteiro, C. C., Inácio, P. R. M., & Freire, M. M. (24). *CloudSim Plus: A cloud computing simulation framework pursuing software engineering principles for improved modularity, extensibility and correctness* (Julio ed., Vol. 2017). IEEE. 10.23919/INM.2017.7987304

Teixeira, A. (n.d.). *BALANCEO DE CARGA*. MUM - MikroTik. Retrieved December 14, 2024, from https://mum.mikrotik.com/presentations/CL16/presentation_3125_1456819785.pdf