

## Renderizado de imágenes 3D mediante una computadora de alto rendimiento Rendering of 3D images through a high-performance computer

C. García-Herrera<sup>a,\*</sup>, R. González-Miranda<sup>a</sup>, J. Sánchez-Espinoza<sup>a</sup>, A. Jaime-González<sup>a</sup>

<sup>a</sup> Ingeniería en Computación, Centro Universitario UAEM Valle de Teotihuacán, Universidad Autónoma del Estado de México, Santo Domingo Azteca, México, México

### Resumen

En el presente documento se analiza el desempeño de una computadora de alto rendimiento y de bajo costo realizando procesamiento de imágenes, usando la técnica de renderizado. Primeramente, se describen trabajos relacionados con la aplicación de dicho tipo de cómputo. En la segunda parte se expone la implementación de la computadora de alto rendimiento considerando principalmente el sistema operativo y el hardware necesario. Acto seguido, se procesan imágenes 3D mediante el renderizado, llevando a cabo tres pruebas fundamentales, en la prueba 1 se procesan las imágenes en el nodo 1, para la segunda prueba las imágenes se procesan en el nodo 1 y en el nodo 3, y para la última prueba se procesan las imágenes en los tres nodos. Finalmente se comprueba que cuando se emplean los tres nodos, el tiempo de procesamiento reduce significativamente en relación a cuando se emplea uno o dos nodos.

*Palabras Clave:* Computo de alto rendimiento, Procesamiento de imágenes, Renderizado, Clúster.

### Abstract

In this document, the compliance of a high performance and low-cost image processing is analyzed, using the rendering technique. First, tasks related to the use of this computer are described. In the second part, the implementation of the high-performance computer is presented, considering the most; the operating system and the necessary hardware. Then, 3D images are processed by rendering, carrying out three fundamental tests. In test 1, the images are processed in node 1, in the second test the images are processed in node 1 and node 3, and in the last test, the images are processed in the three nodes. Finally, it is verified that when the three nodes are used, the processing time is significantly reduced in comparison to when one or two nodes are used.

*Keywords:* High performance computing, Image processing, Rendering, Cluster.

## 1. Introducción

El término cómputo de alto rendimiento o en sus siglas en inglés HPC (*High Performance Computing*), se refiere al hecho de agregar más poder de cómputo a un equipo o conjunto de equipos para ofrecer una capacidad de procesamiento mayor que el ofrecido por las computadoras de uso común (Camazón, 2011). Este tipo de sistemas se ocupa en las áreas, de la ingeniería, investigación médica, exploración modelado, simulación, la manufactura, en el análisis de un gran conjunto de datos (Big Data) y para la inteligencia artificial. Los procesos antes mencionados requieren de una gran cantidad de cálculos computacionales. En la actualidad con la evolución y el crecimiento exponencial de las tecnologías, han aparecido nuevas ciencias y procesos

que demandan un mayor procesamiento de cómputo (Ocampo, 2017), haciendo que computadoras convencionales no sean capaces de satisfacer la demanda (López, 2015), por lo tanto, el motivo de la investigación es la implementación de una computadora de alto rendimiento economizando costos. También se buscará el software que se necesita para el desarrollo del proyecto y por último se realizarán pruebas de procesamiento de imágenes 3D para poder comprobar si los tiempos de renderizado disminuyen. El procesamiento de imágenes es uno de los procesos que más recursos computacionales requiere, razón por la cual, contar con una computadora de alto rendimiento para reducir los tiempos de ejecución es de suma importancia.

\*Autor para la correspondencia: cgarciagh@uaemex.mx

Correo electrónico: cgarciagh@uaemex.mx (Cozobi García Herrera), rodrigoGM1996@hotmail.com (Rodrigo González Miranda), jsancheze@uaemex.mx (Jaqueline Sánchez Espinoza), ajaimmeg@uaemex.mx (Andrés Jaime González).

## 2. Antecedentes

Los sistemas distribuidos y el cómputo de alto rendimiento son especialidades de la computación que iniciaron su formalización en los años ochenta del siglo pasado, dejaron de evolucionar en consecuencia del aumento del poder de cómputo de los equipos que se construían y que recobró su importancia cuando la cantidad de información aumentó y los tiempos de respuesta tuvieron que disminuir.

Un ejemplo de una computadora de alto rendimiento en México es la supercomputadora MITI creada por la UNAM, la cual contribuye a la investigación en los temas de estudio de la estructura del universo, el comportamiento de las partículas y al diseño de nuevos materiales (LANCAD, 2021).

En la Universidad Nacional de Loja se desarrolló un trabajo de investigación acerca del cómputo de alto rendimiento. Se detalla la implementación de un clúster de alto rendimiento ocupado para la programación en paralelo, para el Big Data y para el renderizado de imágenes y videos (Chuquiguanca, 2015).

En la Universidad Nacional José María Argudas se desarrolló un trabajo de investigación acerca de la influencia de un clúster de alto rendimiento para el renderizado de modelos 3D fotorrealistas utilizando la técnica de trazado de rayos con el software POV-Ray (Castro, 2014).

En la actualidad, las imágenes generadas por computadora están presentes en muchos sectores como la investigación, en el sector empresarial, en el desarrollo y el entretenimiento, por mencionar algunos. Estos sectores demandan imágenes de alta calidad para tener una mejor perspectiva del mundo real. Tomando lo anteriormente mencionado, la implementación de una computadora de alto rendimiento ayudará a reducir los tiempos de procesamiento de imágenes y modelos 3D.

Para el caso en particular se investigará cómo desarrollar un prototipo de una computadora de alto rendimiento para emplearla en el renderizado de imágenes 3D.

## 3. Metodología

Para el desarrollo de esta investigación se plantea el objetivo general de implementar una computadora de alto rendimiento que ofrezca un mayor poder de cómputo para el procesamiento de imágenes 3D con el fin de reducir los tiempos de ejecución. De dicho objetivo se desprenden las siguientes actividades:

- Determinar los componentes que conforman la computadora de alto rendimiento.
- Establecer las mejores opciones de software para la gestión de la computadora de alto rendimiento construida.
- Realizar pruebas de rendimiento con imágenes 3D para comprobar si se reducen los tiempos de renderizado.

## 4. Marco teórico

Con la finalidad de abordar la temática del cómputo de alto rendimiento, se tienen que abordar tópicos relacionados a la ciencia de la computación, entre los que destacan el cómputo distribuido, cómputo en paralelo, arquitectura de computadoras y desde luego procesamiento de imágenes.

### 4.1. Computo de alto rendimiento

El computo de alto rendimiento (High Performance Computing) se define como la práctica de agregar poder de computo o de procesamiento a las computadoras a través de diferentes disciplinas de la computación, como la arquitectura de computadoras, sistemas distribuidos y procesamiento en paralelo (Ocampo, 2017). Por medio del uso de supercomputadoras y clúster de computadoras, se resuelven problemas computacionales complejos con un alto desempeño. Algunas aplicaciones que se la da a este tipo de cómputo es en la meteorología, la geología, la investigación nuclear, el análisis de datos (Big Data), el estudio del universo, el tratamiento de imágenes y la aeronáutica, por mencionar algunos (Gómez, 2016).

### 4.2. Computo en paralelo

El cómputo en paralelo es una de las disciplinas que se tiene que considerar en el cómputo de alto rendimiento. La computación o procesamiento en paralelo consiste en acelerar la ejecución de un programa mediante su descomposición en fragmentos que puedan ejecutarse de forma simultánea, cada uno en su propia unidad de procesamiento (Castro, 2014). Dentro del cómputo en paralelo la taxonomía de Flynn clasifica las arquitecturas en paralelo de acuerdo al flujo de datos e instrucciones de una computadora, tal como se describe a continuación (Vargas, 2010):

- Arquitectura SISD (*Single Instruction, Single Data*): Modelo tradicional de computadoras secuenciales, donde una unidad de procesamiento recibe solo una secuencia de instrucciones que opera una secuencia de datos.
- Arquitectura SIMD (*Single Instruction, Multiple Data*): En este tipo de arquitectura existen múltiples procesadores que en sincronía ejecutan la misma secuencia de instrucciones, pero con diferentes datos.
- Arquitectura MISD (*Multiple Instruction, Single Data*): Esta arquitectura se refiere a múltiples instrucciones ejecutando en un único dato, esta arquitectura no es práctica.
- Arquitectura MIMD (*Multiple Instruction, Multiple Data*): Esta arquitectura ejecuta múltiples instrucciones en múltiples datos, donde cada computadora es independiente, pero comparte recursos.

### 4.3 Clúster

Un clúster es un arreglo de computadoras que comparten recursos de hardware y que trabajan de manera paralela y distribuida, con el propósito de funcionar como un único servidor o computadora

(Delgado, 2014). Los clústeres se clasifican en tres categorías de acuerdo con sus características:

- Clúster de alto rendimiento (HPC): Un clúster de alto rendimiento es un arreglo de servidores o computadoras con el propósito de realizar tareas de alta demanda de cálculos, este tipo de clúster es ocupado para la simulación.
- Clúster de alta disponibilidad (HA): Se caracteriza por ser un sistema confiable capaz de analizar, detectar y corregir un gran número de fallos de servicios, de hardware y software, cuando una máquina que conforma el clúster falla, los procesos y la información son migrados a otros nodos del clúster, el propósito de este sistema es mantenerse activo y en funcionamiento la mayor parte del tiempo.
- Clúster de balanceo de carga (LB): Un clúster de balanceo de carga se usa para distribuir la carga de trabajo entre varios procesadores, computadoras, discos u otros dispositivos. Este tipo de sistema es ocupado normalmente en servidores de internet.

### 4.3. Arquitectura de computadoras

La arquitectura de computadoras es un conjunto de ideas fundamentales que se aplican en el diseño o composición de cualquier computadora, desde los más pequeños microprocesadores, hasta la más poderosa supercomputadora (Behrooz, 2005). Existen dos tipos de arquitecturas, la arquitectura Harvard, ocupada por los microcontroladores y procesadores de señales y la arquitectura Von Neumann, ocupada mayormente en los procesadores de las computadoras personales (Montañes, 2017).

La arquitectura de una computadora se compone de diferentes elementos, siendo los más relevantes el procesador dispositivo encargado de toda la lógica, cálculos y donde se procesan todas las instrucciones (Montañes, 2017) La memoria es un componente que aparte del procesador es uno de los más importantes, es la que se encarga de almacenar toda la información y se clasifican en memorias volátiles y no volátiles (Quiroga, 2010). Otros componentes muy importantes son los dispositivos de entrada y salida, mismos que ayudan a los usuarios a tener control del equipo, así como poder ingresar información sin procesar y obtener los resultados de dicho procesamiento (Behrooz, 2005).

### 4.4. Procesamiento de imágenes

El procesamiento digital de imágenes tiene como objetivo principal la manipulación de una o varias imágenes, para lograr como resultado otras que sean de mayor utilidad en donde se extrae la máxima información de las mismas (Charro, 2007). Las imágenes fotorrealistas son imágenes generadas por computadora mediante complejos cálculos matemáticos; este tipo de imágenes en la actualidad tienen un uso en diferentes campos de la ciencia, la simulación y en el entretenimiento (Ordoñez, 2011).

## 4.5. Renderizado

El renderizado es un proceso de alta demanda que pretende generar imágenes fotorrealistas a partir de imágenes 2D y 3D, mismo que se ejecuta por medio de programas informáticos. Este término técnico es utilizado por los animadores o productores audiovisuales (Ordoñez, 2011).

## 5. Implementación

La implementación de la computadora de alto rendimiento, se dividió en dos apartados, en el apartado uno se buscó el software requerido tal como el sistema operativo y aplicaciones necesarias. También en este apartado se llevó a cabo la implementación en máquinas virtuales con el fin de tener un panorama general de lo que se va a desarrollar con las máquinas físicas. En el apartado dos se implementó la computadora de alto rendimiento con computadoras físicas, tomando como base el apartado previo, también se realizaron pruebas de conexión para comprobar que los nodos estén configurados de manera correcta y funcionando. Las características de las computadoras empleadas se detallan en la Tabla 1.

Tabla 1: Características de las computadoras.

Nodo	Marca	Procesador	Memoria RAM	Disco duro
nodoMaestro	Dell	Intel Pentium (R)D	2 GB	160 GB
nodo1	Dell	Intel Pentium (R)D	4 GB	160 GB
nodo2	Lenovo	AMD Ryzen 7	8 GB	1TB
nodo3	Samsung	AMD E1	5.4 GB	2TB

### 5.1. Sistema operativo

El sistema operativo es una parte fundamental para el funcionamiento de una computadora, existen diversos sistemas operativos tales como Windows, MacOS y Linux. A su vez, existen varias distribuciones de este último entre las que destacan: Ubuntu, Debian, CentOS, Red Hat, Fedora, OpenSUSE y Linux Mint, por mencionar algunos (Quiroga, 2010). Estos tipos de sistemas operativos son una buena opción para el desarrollo de la computadora de alto rendimiento, por el grado de personalización y por su costo. Después de analizar los diferentes sistemas, el que se instaló en las computadoras fue Ubuntu en su versión 20.04. Una vez instalado el sistema operativo, se asignan las direcciones IP para cada una de las computadoras con la finalidad de tener comunicación entre ellas, tal como se aprecia en la Tabla 2.

Tabla 2. Direcciones IP de los nodos.

Nodo	Dirección IP	Mascara de red	Puerta de enlace	DNS
nodoMaestro	192.168.0.100	255.255.255.0	192.168.0.1	8.8.8.8
nodo1	192.168.0.101	255.255.255.0	192.168.0.1	8.8.8.8
nodo2	192.168.0.102	255.255.255.0	192.168.0.1	8.8.8.8
nodo3	192.168.0.103	255.255.255.0	192.168.0.1	8.8.8.8

### 5.2. Configuración de la red

Para la configuración de la red se ocupó un switch de la marca TP-Link, modelo TL-SF1005D y cables de red para la comunicación de los nodos. Se configuraron los archivos hosts

en cada equipo, recordando que estos archivos permiten establecer los nombres a cada uno de los nodos del arreglo.

En la Figura 1 se puede observar el archivo del nodo maestro ya configurado.

```

root@cluster: /home/cluster
root@cluster:/home/cluster# cat /etc/hostname
nodoMaestro
root@cluster:/home/cluster# cat /etc/hosts
127.0.0.1    localhost
127.0.1.1    cluster

192.168.0.100  nodoMaestro  nodoMaestro
192.168.0.101  nodo1        nodo1
192.168.0.102  nodo2        nodo2
192.168.0.103  nodo3        nodo3
# The following lines are desirable for IPv6 capable hosts
::1         ip6-localhost ip6-loopback
fe00::0    ip6-localnet
ff00::0    ip6-mcastprefix
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
root@cluster:/home/cluster#

```

Figura 1: Configuración del archivo hosts.

Ya que se configuraron cada uno de los nodos lo que se tiene que hacer es comprobar que tengan comunicación entre sí, para esto, en el nodo maestro se tiene que ejecutar el siguiente comando desde la terminal “*ping ‘nombre del nodo’*”. En la Figura 2 se observa como el nodo maestro tiene comunicación con el nodo 2. Este paso se repite para poder comprobar que exista una comunicación con los demás nodos esclavos.

```

cluster@nodoMaestro: ~
cluster@nodoMaestro:~$ ping nodo2
PING nodo2 (192.168.0.102) 56(84) bytes of data:
64 bytes from nodo2 (192.168.0.102): icmp_seq=1 ttl=64 time=0.171 ms
64 bytes from nodo2 (192.168.0.102): icmp_seq=2 ttl=64 time=0.212 ms
64 bytes from nodo2 (192.168.0.102): icmp_seq=3 ttl=64 time=0.183 ms
64 bytes from nodo2 (192.168.0.102): icmp_seq=4 ttl=64 time=0.186 ms
64 bytes from nodo2 (192.168.0.102): icmp_seq=5 ttl=64 time=0.183 ms
64 bytes from nodo2 (192.168.0.102): icmp_seq=6 ttl=64 time=0.177 ms
64 bytes from nodo2 (192.168.0.102): icmp_seq=7 ttl=64 time=0.175 ms
64 bytes from nodo2 (192.168.0.102): icmp_seq=8 ttl=64 time=0.182 ms
64 bytes from nodo2 (192.168.0.102): icmp_seq=9 ttl=64 time=0.186 ms
64 bytes from nodo2 (192.168.0.102): icmp_seq=10 ttl=64 time=0.180 ms
64 bytes from nodo2 (192.168.0.102): icmp_seq=11 ttl=64 time=0.179 ms
64 bytes from nodo2 (192.168.0.102): icmp_seq=12 ttl=64 time=0.184 ms
64 bytes from nodo2 (192.168.0.102): icmp_seq=13 ttl=64 time=0.184 ms
64 bytes from nodo2 (192.168.0.102): icmp_seq=14 ttl=64 time=0.181 ms
^C
--- nodo2 ping statistics ---
14 packets transmitted, 14 received, 0% packet loss, time 13290ms
rtt min/avg/max/mdev = 0.171/0.183/0.212/0.009 ms
cluster@nodoMaestro:~$

```

Figura 2: Conexión del nodo maestro al nodo 2.

### 5.3. Secure Shell (SSH)

SSH es un protocolo que facilita las comunicaciones seguras ente sistemas usando una arquitectura cliente/servidor, permite conectarse a un host de manera remota a través de un mecanismo de autenticación y las comunicaciones se cifran por medio de claves RSA, sistema criptográfico de clave pública (Chuquiguanca, 2015). Este protocolo se tiene que instalar en los nodos del arreglo y se tiene que comprobar que exista una conexión entre sí.

En la Figura 3 se observa cómo el nodo maestro tiene acceso de manera remota hacia el nodo 3, este paso se repite para cada una de las computadoras, para comprobar que el nodo maestro tenga acceso.

```

cluster@nodo3: ~
cluster@nodoMaestro:~$ ssh nodo3
cluster@nodo3's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-46-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Se pueden aplicar 14 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales ejecute: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Thu Jan 27 10:21:32 2022 from 192.168.0.100
cluster@nodo3:~$

```

Figura 3: Conexión por medio SSH del nodo maestro al nodo 3.

### 5.4. Network File System (NFS)

NFS es un protocolo de aplicación del modelo OSI, es utilizado para sistemas de archivos distribuidos en un entorno de red de computadoras de área local, permite que distintos sistemas tengan acceso a los archivos con un modelo cliente servidor, este protocolo es utilizado para que los nodos esclavos tengan acceso al sistema de directorios del nodo maestro (Chuquiguanca, 2015), En la Figura 4 se observa el protocolo NFS compartiendo la carpeta *home* del nodo maestro, hacia el nodo 1.

```

cluster@nodoMaestro: ~
cluster@nodoMaestro:~$ ssh nodo1 ls -l
cluster@nodo1's password:
total 36
drwxr-xr-x 5 cluster cluster 4096 sep 11 2017 blender-2.79
drwxr-xr-x 2 cluster cluster 4096 ene 19 13:24 Descargas
drwxr-xr-x 2 cluster cluster 4096 dic 17 16:56 Documentos
drwxr-xr-x 2 cluster cluster 4096 dic 17 16:56 Escritorio
drwxr-xr-x 5 cluster cluster 4096 ene 27 10:19 Imágenes
drwxr-xr-x 2 cluster cluster 4096 dic 17 16:56 Música
drwxr-xr-x 2 cluster cluster 4096 dic 17 16:56 Plantillas
drwxr-xr-x 2 cluster cluster 4096 dic 17 16:56 Público
drwxr-xr-x 2 cluster cluster 4096 dic 17 16:56 Videos
cluster@nodoMaestro:~$

```

Figura 4: Protocolo NFS activo en el nodo 1.

### 5.5. Ganglia Monitoring System

Este sistema de monitoreo está orientado al cómputo de alto rendimiento, mide el desempeño del CPU, la memoria RAM y la red de los nodos del arreglo. También se pueden agregar más métricas de medición si es necesario. Ganglia ocupa XML para la representación de los datos. XDR (External Data Representation) es ocupado para el análisis del tráfico de la red y RRDtool (Round Robin Database Tool) es ocupado para el registro y graficado de los datos capturados (Ortega, 2014).

En la Figura 5 se observa la interface de Ganglia en donde muestra el número de nodos conectados, el total de CPU's, la cantidad de memoria RAM y la cantidad de memoria de almacenamiento.

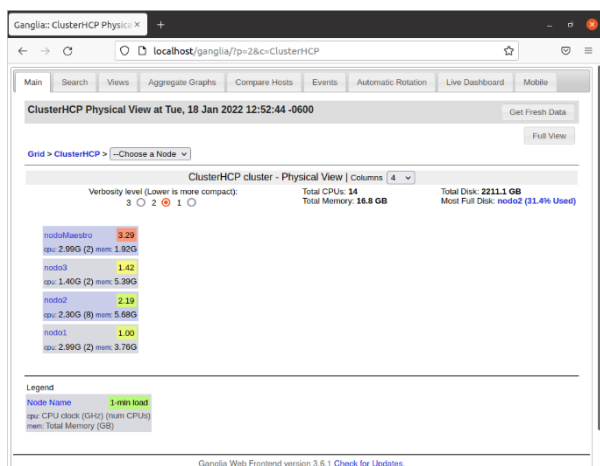


Figura 5: Interface gráfica de Ganglia.

### 5.6. Blender

Blender es el conjunto de creación 3D de código abierto con el modelado de aparejo, animación, simulación, renderizado, composición y edición de video. Otra característica importante es que es multiplataforma, su interface utiliza OpenGL. La versión que se utilizó en este trabajo fue la versión 2.79 debido a que dicha versión proporciona las herramientas para poder configurar una granja de renderizado (Pérez, 2011). En la Figura 6 se puede observar la interface gráfica de Blender.

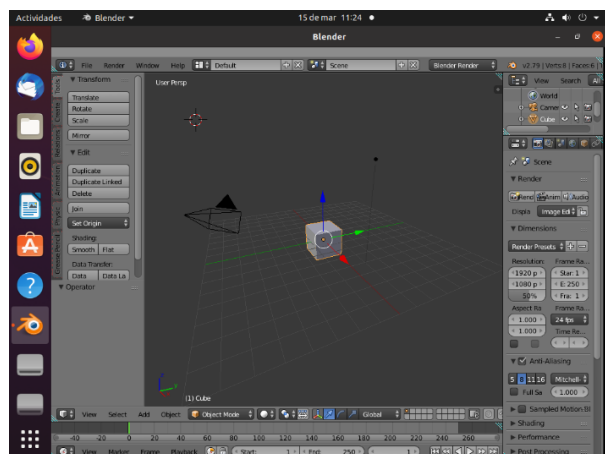


Figura 6: Interface gráfica de Blender.

## 6. Pruebas de renderizado

Ya que se configuraron los nodos, el siguiente paso es realizar las pruebas de renderizado, para esto se emplearon tres imágenes 3D. El software Blender cuenta con su propio motor de renderizado, llamado blender render capaz de renderizar en tiempo real. Las tres imágenes 3D se renderizaron con este motor activando la opción network render; esta opción que se encuentra activa por defecto permite realizar y monitorear el renderizado en un ambiente distribuido.

En la prueba 1, se renderizaron las imágenes 3D en el nodo 1, para la segunda prueba las imágenes 3D se renderizaron en el nodo 1 y en el **nodo 2** y para la última prueba se ocuparon todos los nodos del arreglo.

### 6.1. Pruebas de la imagen 3D 1

Para estas pruebas se ocupó una imagen 3D sencilla. En la Figura 7 se puede observar dicha imagen ya abierta en el software Blender, antes del renderizado.

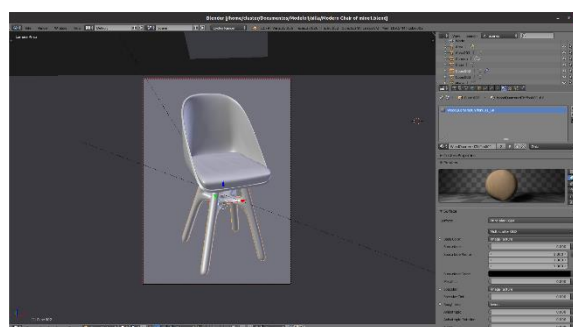


Figura 7: Imagen 3D 1 sin renderizar.

Durante la ejecución de las pruebas de la imagen 3D 1 se obtuvieron diferentes tiempos de renderizado, como se puede apreciar, los tiempos varían a medida que se agregan nodos a la computadora de alto rendimiento. En la Tabla 3 se pueden observar los tiempos de renderizado. En la figura 8 se muestra el promedio de cada prueba.

Tabla 3: Resultados de la imagen 3D 1.

Renderizado	Prueba 1	Prueba 2	Prueba 3
	Nodo1	Nodos1,2	Nodos1,2,3
Renderizado1	16:10 min	10:10 min	3:27 min
Renderizado2	16:06 min	10:07 min	3:21 min
Renderizado3	16:14 min	10:10 min	3:23 min

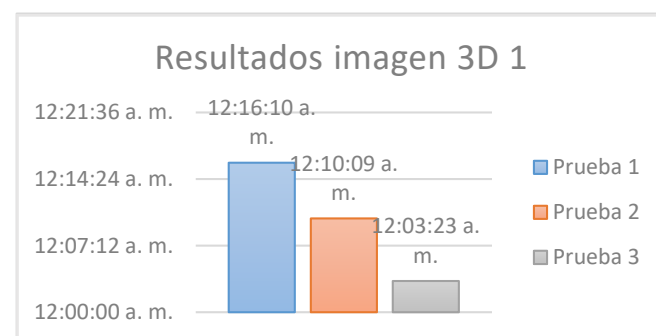


Figura 8: Promedio de los resultados de la imagen 3D 1.

### 6.2. Pruebas de la imagen 3D 2

Para estas pruebas se ocupó una imagen 3D más compleja, en este caso se renderizó una serie de figuras geométricas, con la finalidad de poder generar más carga de trabajo a los nodos. De la misma forma que en la primera prueba, la imagen 3D se abrió en el software Blender. En la Figura 9 se observa la imagen sin renderizar.

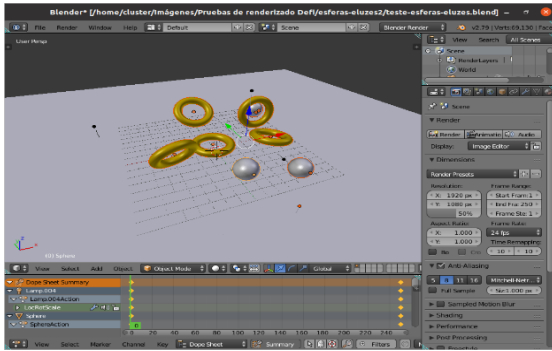


Figura 9: Imagen 3D 2 sin renderizar.

Las pruebas se repiten para la imagen 3D 2, en estas pruebas se observó que los tiempos de procesamiento se reducen a medida que se agregan nodos a la computadora de alto rendimiento. En la Tabla 4 se observan los tiempos de renderizado de la imagen en cuestión y en la Figura 10 se observan los promedios de las pruebas.

Tabla 4: Resultados de la imagen 3D 2.

Renderizado	Prueba 1	Prueba 2	Prueba 3
	Nodo1	Nodos 1,2	Nodos 1,2,3
Renderizado1	36:13 min	22:09 min	5:41 min
Renderizado2	36:06 min	21:48 min	6:22 min
Renderizado3	36:08 min	21:58 min	5:38 min

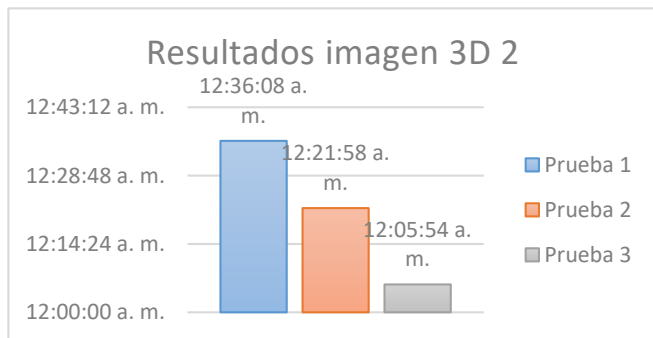


Figura 10: Promedio de los resultados de la imagen 3D 2.

### 6.3. Pruebas de la imagen 3D 3

La imagen 3D 3 es una imagen más detallada de forma que al procesarla se llevará más tiempo. En la Figura 11 se observa la imagen 3D 3 sin renderizar.

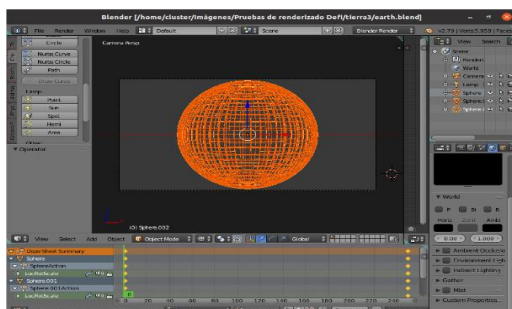


Figura 11: Imagen 3D 3 sin renderizar.

Se realizaron las mismas pruebas que en las imágenes anteriores, también se observa una reducción en los tiempos de ejecución de los renderizados. En la Tabla 5 se observan los tiempos de renderizado y en la Figura 12 se observan los promedios de las pruebas.

Tabla 5: Resultados de la imagen 3D 3.

Renderizado	Prueba 1	Prueba 2	Prueba 3
	Nodo1	Nodos 1,2	Nodos 1,2,3
Renderizado1	37:17 min	27:06 min	6:49 min
Renderizado2	37:33 min	24:04 min	6:51 min
Renderizado3	37:27 min	23:37 min	6:49 min

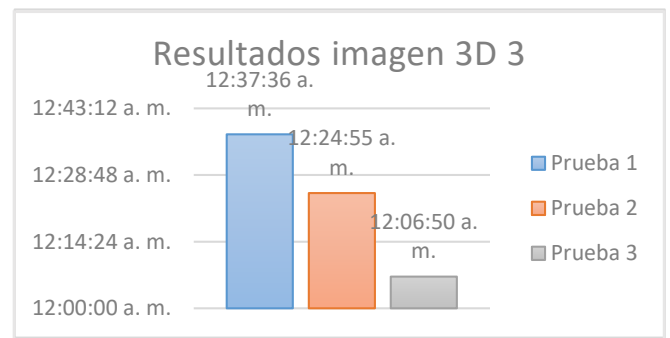


Figura 12: Promedio de los resultados de la imagen 3D 3.

### 6.4. Resultados de los renderizados

Para las pruebas de renderizado se configura el Blender de manera que sólo se procesen 100 fotogramas de cada imagen 3D, tomando como referencia una cámara que permite tomar diferentes ángulos para dar fluidez a la imagen 3D. Durante las pruebas realizadas se comprobó que conforme se agregan nodos a la computadora de alto rendimiento, los tiempos de renderizado se disminuyen. Si se tiene una computadora de este tipo y se tiene que renderizar un conjunto de imágenes, el tiempo que se tomará para dicha actividad será menor en comparación del tiempo que tomaría realizarlo con un solo equipo. En la Figura 13 se puede observar la imagen 3D 1 renderizada, en la Figura 14 la imagen 3D 2 y en la Figura 15 la imagen 3D 3.



Figura 13: Resultados de la imagen 3D 1.

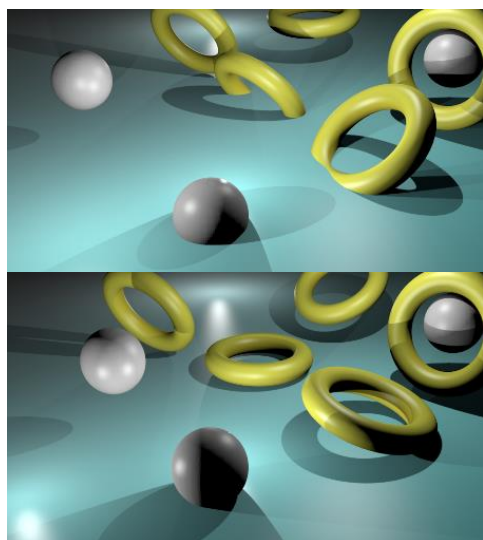


Figura 14: Resultados de la imagen 3D 2.



Figura 15: Resultados de la imagen 3D 3.

## 7. Conclusiones

La computadora de alto rendimiento básicamente está conformada por cuatro computadoras o nodos con arquitecturas de hardware muy limitadas que ya no se encontraban en funcionamiento y que no fue necesario adquirir, compartiendo recursos a través de una red de datos para llevar a cabo el procesamiento de información. El sistema operativo seleccionado es Linux, en particular la distribución Ubuntu en su versión 20.04 de código abierto. Para la administración de la computadora se instaló SSH (Secure Shell), NFS (Network File System) y Ganglia Monitor System. Blender fue el software encargado de llevar a cabo el renderizado de las imágenes 3D asignando la carga a los nodos de la computadora en cuestión.

El objetivo central del presente documento fue renderizar imágenes 3D mediante el empleo de una computadora de alto rendimiento y comprobar si los tiempos de ejecución

disminuyen al incorporar más nodos de procesamiento. Efectivamente, durante las pruebas se observó una reducción de los tiempos de renderizado ya que en la prueba en la que se utilizan los tres nodos para dicho renderizado, los tiempos se reducen drásticamente en relación a cuando solamente se emplea un nodo.

Resulta conveniente enfatizar que de acuerdo a las características actuales de los sistemas operativos y software para renderizado, las computadoras empleadas no podrían realizar dicha actividad si trabajaran en forma individual, ya que no cumplen con los requisitos de hardware para soportar dichas aplicaciones. En el momento de construir la computadora de alto rendimiento, se habilitan de nueva cuenta los equipos y se tiene la capacidad de emplear software aún disponible que permite llevar a cabo el renderizado de las imágenes 3D.

## 8. Referencias

- Behrooz, P. (2005). *Computer Architecture: From Microprocessors To Supercomputers*. Santa Barbara: McGraw-Hill.
- Bolaños, M. (2014). Computación de alto rendimiento para el procesamiento de imágenes digitales. *Ventana Informática*, 56-65.
- Camacho, R. (2012). *Computo Integrado*. Obtenido de Blogspot: <http://rcmcomputointegrado.blogspot.com/2012/04/arquitectura-von-neumann.html>
- Camazón, J. (2011). *Sistemas Operativos Monopuesto*. Madrid: EDITEX.
- Carpena, A. (2017). Configuración y Testeo de un Clúster GPU. (*Tesis de Grado*). Universidad de Alicante, Alicante.
- Castro, B. (2014). Influencia de un Clúster de Computadoras de Alto Rendimiento en el Tiempo de Renderización de Modelos 3D Fotorealistas. (*Título de grado*). Universidad Nacional José María Arguedas, Andahuaylas.
- Charro, C. (2007). Modelo tridimensional De La Historia Geológica Del Volcán Cotopaxi. (*Tesis de Licenciatura*). Escuela Politécnica Nacional, Ecuador.
- Chuquiguanca, L. (2015). Arquitectura cluster de alto rendimiento utilizando herramientas de software libre. (*Tesis de licenciatura*). Universidad nacional de Loja, Ecuador.
- Delgado, J. (2014). Desarrollo de un Prototipo Virtualizado de Computación de Alto Desempeño (HPC) Para la Analítica de Datos. (*Título de licenciatura*). Institución Universitaria Politécnico Grancolombia, Bogotá.
- Gómez, Y. (2016). Implementación del Método PIC Sobre un Clúster con GPU Usando CUDA. (*Tesis de Maestría*). Universidad Tecnológica de Pereira, Pereira.
- González, C. (2007). Optimización del proceso de render 3D distribuido con software libre. *Revista de la Asociación de Técnicos de Informática*(190).
- Hoeger, H. (2004). Introducción a la Computación Paralela. (*Tesina para Licenciatura*). Centro Nacional de Cálculo Científico Universidad de Los Andes, Mérida.
- LANCAD. (17 de Febrero de 2021). *MIZTLE*. Obtenido de Laboratorio Nacional de Cómuto de Alto Desempeño: <http://www.lancad.mx/?p=59>
- López, F. (2015). *Sistemas Distribuidos*. Cuajimalpa: Universidad Autónoma Metropolitana.
- Montañes, J. (2017). Sistema Basado en microcontrolador para la Automatización de un acuario. (*Título de Mástría*). Universitat Politècnica de València, València.
- Ocampo, I. (2017). *Introducción a la Supercomputación en Perú*. Perú: Depósito Legal en la Biblioteca Nacional del Perú.
- Ordóñez, O. (2011). Clúster de renderizado para la generación de modelos tridimensionales. (*Tesis De Maestría*). Universidad TECMilenio Del Sistema Tecnológico De Monterrey, Morelia.
- Ortega, C. (2014). Diseño y elaboración de un clúster HPC: Software de sistema. (*Tesis de licenciatura*). Universidad politécnica de cataluña (UPC), Barcelona.
- Párraga, V. (2017). Desarrollo de un Algoritmo que Permita Paralelizar Imágenes 3D Sobre Tecnología NVIDIA y OPENCL. (*Tesis de licenciatura*). Universidad Politécnica Salesiana Sede Quito, Quito.
- Quiroga, P. (2010). *Arquitectura de computadoras*. Buenos Aires: Alfaomega.

- Rojas, M. (2015). Implementación De Un Sistema De Clúster Con Rocks En Una Red Tradicional. (*Tesis de grado*). Universidad Tecnológica Equinoccial, Ecuador.
- Ronán, H. (2011). Solución De Problemas De Optimización Mediante Algoritmos Genéticos Aplicado Cómputo De Alto Rendimiento. (*Tesis De Mastría*). Instituto Politécnico Nacional, Tijuana.
- Solorzano, F. (2010). GNU/Linux Desde la construcción social de la tecnología. *Reflexiones*, 89, 87-95.
- Pérez, P. (2011). *Manual de modelado y animación con Blender*. Alicante: Universidad de Alicante.
- Tanenbaum, A. (1996). *Sistemas Operativos Distribuidos*. Prentice Hall.
- Vargas, J. (2010). Cálculo de Estructuras Utilizando Elemento Finito con Cómputo en Paralelo. (*Tesis para Maestría*). Centro de Investigación en Matemáticas A.C., Guanajuato.