GUÍA DE IMPLEMENTACIÓN DE UN LABORATORIO PARA LA REALIZACIÓN DE PRUEBAS DE SOFTWARE DISTRIBUIDO Y PROCESAMIENTO EN PARALELO

JOSE LUIS VASQUEZ GIRALDO

UNIVERSIDAD CATOLICA DE COLOMBIA
FACULTAD DE INGENIERIA
PROGRAMA DE INGENIERÍA DE SISTEMAS
BOGOTÁ D.C. 2014

GUÍA DE IMPLEMENTACIÓN DE UN LABORATORIO PARA LA REALIZACIÓN DE PRUEBAS DE SOFTWARE DISTRIBUIDO Y PROCESAMIENTO EN PARALELO

JOSE LUIS VASQUEZ GIRALDO

TRABAJO DE GRADO
MODALIDAD "TRABAJO DE INVESTIGACIÓN SIN DESARROLLO"

Director
Holman Diego Bolívar Barón
Ingeniero de Sistemas

UNIVERSIDAD CATOLICA DE COLOMBIA
FACULTAD DE INGENIERIA
PROGRAMA DE INGENIERÍA DE SISTEMAS
BOGOTÁ D.C. 2014

Nota de aceptación

Aprobado por el comité de grado en cumplimiento de los requisitos exigidos por la Facultad de Ingeniería y la Universidad Católica de Colombia para optar al título de ingenieros de Sistemas.

_____

Ingeniero Holman Diego Bolívar
Director

_____

Ingeniero Carlos Pulido
Revisor Metodológico

Bogotá D. C. Mayo 29 de 2014

AGRADECIMIENTOS

Agradezco a mis padres Jose Luis y Lucia por darme la vida y creer en mí, a mi esposa Tulika por su apoyo incondicional en esta etapa de mi vida, a mis suegros Neelie y BN por ver en mi un futuro brillante.

Gracias por brindarme la tranquilidad requerida en los momentos más difíciles.

El presente trabajo de investigación fue realizado bajo la supervisión del Ingeniero Holman Bolívar a quien quiero agradecer su paciencia, tiempo y dedicación que tuvo para que esto saliera de forma exitosa.

Gracias por su apoyo por ayudarme a sacar este proyecto adelante.

CONTENIDO

Pág.

LISTA DE TABLAS

LISTA DE FIGURAS

Pág.

LISTA DE ANEXOS

Pág.

GLOSARIO

App: “Application software”. Es el término utilizado comúnmente como abreviatura de aplicación informática, la cual se define como un programa informático diseñado para ayudar al usuario a realizar una serie de tareas específicas. Además, las aplicaciones pueden ser estándar o desarrolladas a medida para cubrir las necesidades particulares de un usuario en concreto.

Backup: Copia de seguridad que se realiza sobre ficheros o aplicaciones contenidas en un ordenador con la finalidad de recuperar los datos en el caso de que el sistema de información sufra daños o pérdidas accidentales de los datos almacenados. Los dispositivos más empleados para llevar a cabo la técnica de backup pueden ser cintas magnéticas, DVD, discos duros, discos ópticos, USB o hasta incluso la implementación de un servicio remoto de copia de seguridad.

Blender 3D: es un programa informático multiplataforma, dedicado especialmente al modelado, animación y creación de gráficos tridimensionales.

Cloud Computing: Es un nuevo concepto tecnológico que se basa en que las aplicaciones software y los equipos hardware con capacidad de proceso y almacenaje de datos no están en el PC o equipos del usuario, sino que están ubicado en un Datacenter que permite a los usuarios acceder a las aplicaciones y servicios disponibles a través de Internet o como se conoce coloquialmente a través “la Nube” de Internet, de una forma sencilla y cómoda.

Clúster: Conjunto de servidores que trabajan como una única maquina mejorando el desempeño de las transacciones y operaciones implantadas en este sistema.

Data Center: Un centro de almacenaje de datos y que provee servicios de negocio que entrega de forma segura aplicaciones y datos a usuarios remotos a través de Internet.

End to End: extremo a extremo, se trata de soluciones cloud basadas en el principio del end-to-end, el cual establece que las funciones específicas de las aplicaciones deben residir en el host final de una red y no en los nodos intermedios, siempre y cuando puedan ser implementadas completa y correctamente en dicho host final.

Grid Computing: Tecnología que permite la coordinación de todo tipo de recursos heterogéneos (cómputo, almacenamiento, aplicaciones, etc., de diferentes arquitecturas), trabajando de forma descentralizada. Supone el uso integrado de equipamiento de alto rendimiento, redes, y bases de datos ubicadas en distintas

instituciones. Suele utilizarse este modelo por universidades o laboratorios de investigación, que se asocian, obteniendo así resultados sinérgicos.

IaaS: “Infraestracture as a Service” o “Infraestructura como Servicio”. Con una Infraestructura como servicio (Iaas) lo que se tiene es una solución basada en virtualización en la que se paga por consumo de recursos: espacio en disco utilizado, tiempo de CPU, espacio en base de datos y transferencia de datos.

KVM: Virtualización por Kernel (Kernel-Bases Virtual Machine) es una infraestructura de virtualización para el kernel de Linux que lo convierte en un Hypervisor para de esta forma llevar a cabo virtualización de alto nivel.

LAN: Local Área Network, En redes significan redes de área local formadas por equipos de cómputos conectados entre sí.

Linux: Linux es un sistema operativo: un conjunto de programas que le permiten interactuar con su ordenador y ejecutar otros programas.

Mainframe: Computadora de gran capacidad de cómputo y costosa, utilizada principalmente en empresas que necesitan procesar gran cantidad de datos o soportar gran cantidad de usuarios. Puede funcionar durante largos períodos de tiempo sin ninguna interrupción, pudiéndose reparar en funcionamiento.

Máquina Virtual: Ordenador que está construido utilizando recursos virtualizados. Este sistema se comporta a nivel lógico de manera idéntica a la de un ordenador físico, de modo que el Sistema Operativo o aplicaciones que corren sobre él no detectan la diferencia.

Multiblend: Es un conjunto de scripts que permiten la renderización en paralelo usando una red de computadores que aumentara de forma dramática la velocidad de renderizado.

Multitenancy: Uso común entre todos los clientes y usuarios de los servicios de computación en la nube desde la misma plataforma tecnológica del proveedor contratado.
On-demand: Término referido al concepto de —bajo demanda. Dentro del ámbito tecnológico se utiliza para expresar la flexibilidad de los productos cloud, basados en un modelo de pago por uso y en los cuales el proveedor pone a disposición del cliente todos sus recursos, pudiéndolos usar bajo petición previa.

On-premise: Modelo referido al esquema tradicional de licenciamiento, es decir la empresa adquiere las licencias que le otorgan derecho de uso de los sistemas del proveedor, los integra en sus propias instalaciones y mantiene sus datos dentro de su propia infraestructura de tecnología.

Open Source: El software libre no debe ser confundido con el software gratuito o freeware. El software libre no tiene por qué ser gratuito, sino que adquiere su denominación por el hecho de que el código fuente es “Código Abierto” (Open Source). Los programas bajo licencia GPL (“General Public License”), una vez adquiridos, pueden ser usados, copiados, modificados y redistribuidos libremente, salvo determinados casos en los que se indiquen ciertas restricciones, como la obligación de distribuir el software con la misma licencia.

PaaS: “Platform as a Service” o “Plataforma como Servicio”. Es el resultado de la aplicación al desarrollo de Software del modelo SaaS . El modelo PaaS abarca el ciclo completo para desarrollar e implantar aplicaciones desde Internet.

Php: PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

Python: lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

SaaS: “Software as a Service” o “Software como Servicio”. Es aquella aplicación ofrecida por su creador a través de Internet para su utilización por varios clientes manteniendo la privacidad de sus datos y la personalización de la aplicación.

Storage: En un computador, el storage es el lugar donde los datos son guardados para acceder a ellos de forma electromagnética u óptica por el procesador de la computadora.

Virtualización: Es el concepto que describe cómo en un solo computador físico se coordina el uso de los recursos para que varios sistemas operativos puedan funcionar al mismo tiempo de forma independiente y sin que ellos (los SO) sepan que están compartiendo recursos con otros sistemas operativos.

Virtualbox: Es el nombre que lleva el software de virtualización de computadores (Virtual Machine), Se trata de un sistema de virtualización por software, en el que se emula un sistema físico (Computador) con unas características de hardware determinadas. Virtualbox permite ejecutar varios sistemas operativos de forma independiente sobre una infraestructura física.

VMWare: Es el nombre que lleva el software de virtualización de computadores (Virtual Machine), Se trata de un sistema de virtualización por software de tipo propietario.

WAN: Wide Área Network. La interconexión de equipos a nivel corporativo entre 2 puntos geográficos se considera como una red WAN.

RESUMEN

Es posible hacer un mejor uso de recursos tecnológicos llevando a cabo un análisis de la situación real de una organización, en la cual los ambientes de investigación sean un elemento que se tiene en cuenta en el desarrollo institucional. La guía de implementación de un laboratorio Grid para renderización en paralelo se realizó pensando en necesidades académicas en la cuales los estudiantes entiendan la necesidad de la computación paralela al igual que se observe cómo es posible maximizar el uso de los recursos que se posean en el momento.

La guía se elaboró buscando la mejor forma de explicar el una posible solución a los problemas computacionales, apoyándose completamente en software libre y excluyendo que sea necesario tener ciertos privilegios institucionales. Es decir se puede implementar en cualquier casa o laboratorio universitario.

Esta guía concluye que si es posible optimizar tareas que tienen uso intensivo de memoria y procesador al virtualizar por núcleo y paralelizar las tareas necesarias. Estas labores pueden ser dramáticamente incrementadas y de la misma forma se obtiene un límite de la optimización ya que el tiempo ahorrado con grandes cantidades de nodos puede no ser representativo según el objeto de la tarea.

Al virtualizar por cantidad de núcleos se obtuvo un rendimiento alto, dejando en claro que es posible construir un súper computador que funcione en horas valle institucionales, o simplemente optimizar el rendimiento un mismo host como cualquier equipo de computo ya sea de casa u oficina. Se pudo observar que en los laboratorios de la Universidad Católica de Colombia, fue posible implementar un supercomputador para renderizar modelos hechos en Blender3d.

De la forma en la cual fue desarrollada la guía, la implementación se sale de las restricciones institucionales que se puedan tener, y así no se tenga infraestructura de red dedicada para el laboratorio, es posible hacer en paralelo el renderizado de modelos 3d y de la misma forma permitir que quien lleve a cabo esta guía, pueda comprender el problema, entender el escenario y aplicar una solución que va a obedecen a una relación para optimización en donde más equipos es igual a menos tiempo y a un principio de desarrollo llamado divide y vencerás.

Palabras Clave: Tecnología de Información, Modelos, Ciencia Computacional, Experimentos Científicos.

ABSTRACT

It is possible to make better use of technological resources bringing an analysis of one organization´s current scenario. Research environments are the key elements while thinking about institutional development. The Grid-Like parallel rendering guide was created considering the academic requirements that students must understand as they observe how it´s possible to maximize the actual resources held in the institution.

The guide was developed seeking the best way to explain a possible solution to computational problems, relying entirely on free software and excluding necessary institutional privileges. It can be implemented in any home or college laboratory.

This guide concludes that it is possible to optimize are memory intensive tasks and to core virtualization will allow to parallelize the necessary tasks. These efforts may dramatically increase the results and also it lets detect that optimization has a boundary, there will be a point where the optimization will be minimal and it won't be representative for the core business.

Applying processor core virtualization to parallel processing, show as results a very high gaining of computational power performance, making clear that is possible to build a supercomputer using the valley hours free computing resources, or simply optimize performance the same host as any computer found at home or an office. It was observed in the labs of the Universidad Católica de Colombia that it was possible to implement a supercomputer to render Blender3D models.

The way the guide was developed released, the students implementing it will be away of institutional constraints such as network infrastructure, It´s possible to render 3D models using parallel computing regardless of the institutional limitations as long as there is one network and computers connected to it. Whoever follows this guide will be able to understand the problem, the scenario and implement a solution that will obey optimization principles where more computing devices are equals to less time and will obey another development principle named "divide and conquer"

Keywords: Information Technology, Models, Computer science, Science experiments.

INTRODUCCIÓN

La capacidad de cómputo y el análisis de datos se pueden ver como una poderosa herramienta para la toma de decisiones; la cantidad de datos analizados puede influir en la toma de decisión. Este análisis de datos se ve en limitador por el poder de cómputo necesario en un tiempo determinado, es aquí donde la capacidad de cómputo juega un papel importante en el análisis en bruto que dé resultados útiles.

Mencionando un popular ejemplo de la relación que hay entre la capacidad de cómputo y el procesamiento de datos, es posible señalar el Bosón de Higgs[1], después del experimento que sustentó su información científica, se tuvo que esperar alrededor de 8 meses para analizar los datos para así publicar la información ya probada.

Es necesario aplicar conceptos como lo son el computador paralelo que es capaz de ejecutar varias instrucciones simultáneamente. Sumado a este es posible hacer uso de la cComputación paralela en la cual habrá varios procesadores trabajando juntos para resolver una tarea común. La meta general es que cada procesador trabaje en una parte del problema. Los procesos pueden intercambiar datosmediante una interconexión de red.

Tomando la ventaja de esquemas paralelos, es posible a través del uso de multi procesadores en un mismo oquipo o en varios equipos consolidar por demanda las unidades de cómputo necesarias para procesar segmentos de una tarea global. Para los efectos de esta investigación se ha abordado el tema desde la virtualización por núcleo, es decir, hacer un bypass del SO instalado y a través de tecnologías de virtualización hacer que un computador se comporte realmente como varios computadores.

Es relevante mencionar que se espera más velocidad en ejecución una tarea global al reducir tiempos de cómputo, de esta forma es posible obtener una buena relación de costo/prestaciones. Este acercamiento le permitirá al investigador ver la escalabilidad de hardware en la que los diseños pueden ser más eficientes y flexibles.

Teniendo en cuento la importancia de la capacidad de cómputo para analizar datos, esta investigación apunta a despertar el poder computacional que se

[1] GUNION, J. F. DAWSON The Higgs hunter's guide. En. H. E. HABER, G. KANE AND S. *et al*. The Mega Guide. No. 2. (Jul 1990).

encuentra oculto en los equipos institucionales para permitir avanzar en otras áreas de investigación.

El alcance de la computación paralela tiene impacto en una gran variedad de areas. Es posible ver como en aplicaciones criticas de ingeniería y diseño como el diseño de aviones, circuitos de alta velocidad requieren gran análisis de datos.

Para que estos modelos salgan de un campo teorico de un aula de clase por motivos tecnológicos en una institución, es necesario abordar la ivnestigacion con modelos simples y realistas, en los que existan nodos secuenciales intercomunicados a través de una red. Los costos de este despliegue deben ser bajos y el acceso a estos equipos total.

1. PLANTEAMIENTO DEL PROBLEMA

El poder computacional actual se está quedando relegado debido a que no es posible aumentar la potencia de un procesador de forma ilimitada, "los fabricantes de CPUs están aumentando la cantidad de núcleos por procesador para elevar el poder computacional"[2], aún así el procesamiento de tareas más complejas como lo son el análisis de datos estadísticos globales o la renderización de imágenes 3D ya sea para el estudios de partículas o elaboración de películas. De esta forma es posible expresar el problema del tiempo computacional de la siguiente forma.

Una película como Transformers que se presenta a 24 FPS[3] (Frames per second) para un total de 1440 Frames por minuto. Teniendo en cuenta que esta misma película dura 154 minutos obtenemos un número aproximado de 221760 frames en la película. Esta producción tomo al rededor de 72 horas de renderización por frame debido a las complejas y detalladas escenas. Si multiplicamos 221760 FPS por 72 Horas que toma renderizar cada frame, la creación del archivo de video le tomaría alrededor de 15966720 Horas que representan unos 1822 años aproximadamente. Los datos de esta película han sido tomados del sitio web del productor Michael Bay[4].

El tiempo del ejemplo anterior ilustra claramente el problema a abordar y el por qué es esencial entrar en el campo e la computación distribuida y procesamiento en paralelo.

Este proyecto apunta a generar el conocimiento requerido para usar software distribuido en tareas extenuantes como lo es la renderización de animaciones. Con la investigación actual se busca reducir el tiempo computacional requerido para los fines requeridos.

¿Es posible optimizar los procesos de renderizado en el mismo hardware para obtener un mejor desempeño de las mismas aplicaciones de forma local y/o escalable?.

---

[2] TESLA, N. Parallel Programming and Problem Solving with CUDA. [en línea] <https://www.youtube.com/watch?v=olvtFAArb9I>[citado en Marzo 21 de 2014]..

[3] BRADNER, S. Benchmarking terminology for network interconnection devices. En: Revista Benchmarking. No. 12 (Feb., 1991).

[4] BAY, M. Transformers Revenge of the Fallen Fun Facts. [en línea]. <http://michaelbay.com/2009/06/17/transformers-revenge-of-the-fallen-fun-facts/> [citado en 21 de marzo de 2014] .

2. OBJETIVOS DEL PROYECTO

2.1 OBJETIVO GENERAL

Desarrollar una guía referente a la implementación de un laboratorio de Grid para la realización de pruebas de software distribuido y procesamiento en paralelo.

2.2 OBJETIVOS ESPECÍFICOS

- Validar las capacidades tecnológicas para un laboratorio de Grid en la sala de informática de la facultad de Ingeniería de acuerdo a los requerimientos de investigación de la facultad de Ingeniería.
- Implementar un servicio Grid, de acuerdo a los requerimientos de investigación y capacidades tecnológicas de la sala de informática de la facultad de ingeniería.
- Desarrollar una guía de usuario y configuración del laboratorio de Grid propuesto.

3. MARCO REFERENCIAL

3.1 MARCO CONCEPTUAL

El concepto de computación distribuida hace referencia a las redes de comunicaciones entre equipos de cómputo. "La tecnología Grid fue propuesta por Ian Foster y Carl Kesserman"[5] a mediados de los 90s y luego se adaptó en el área informática. Esta tecnología busca el acceso remoto de recursos computacionales como lo son recursos de procesamiento, recursos de almacenamiento o el manejo de aplicaciones particulares con el fin de incrementar el poder de cálculo para ser administrado y distribuido y así poder llegar a la realización de cálculos avanzados.

La Grid computacional se diferencia de los demás sistemas cliente servidor o tecnologías como (CORBA o .NET)[6] en su orientación a los recursos computacionales más no a la información. La seguridad está presente todo el tiempo y el tipo de comunicación que maneja es asíncrono. Esta tecnología depende del ancho de banda pero a su vez brinda un alto nivel de procesamiento, permitiendo la maximización de recursos que pueden ser explotados en todos los sectores que requieren estudio de datos.

La tecnología de computación distribuida brinda beneficios como:

- Aumento de poder de computo por menor inversión al usar equipos existentes.
- No requiere una infraestructura de última generación para funcionar.
- No es necesario pensar en licenciamiento para implementarse debido a que la mayoría del software Grid es libre.
- Maximiza la inversión presente de la institución.
- Se obtiene un supercomputador.

Hay que tener en cuenta las tres maneras para usar los recursos disponibles en los equipos debido a que no todos tienen las mismas características, pueden ser de diferente arquitectura, procesamiento, SO, Software, etc.,. Estas maneras son: 1. Ejecutar una aplicación en un equipo disponible en la Grid, 2. Diseñar una aplicación para dividir el trabajo en la Grid y así obtener la maximización de

[5] FOSTER, I. The Grid 2: Blueprint for a new computing infrastructure. En. C. KESSELMAN *et al*. New Computing Infrastructure (Jan 2003) P 23-24.

[6] KHAN, S.Performance comparison of ICE, En. International Journal of Computer Applications HORB (Ago 2010) P 120.

rendimiento. 3. Ejecutar una aplicación varias veces en diferentes equipos de la Grid.
La capacidad de almacenamiento en el Grid tiende a hacer redundante, es decir crea copias de los programas que se ejecutan para acceder más rápido a ellos mismos, de esta manera se incrementa la fiabilidad y el rendimiento.

La computación distribuida presenta ventajas y desventajas descritas a continuación.

Ventajas:

- Integración de varios equipos sin importar su arquitectura.
- Tolerancia contra fallas.
- Acceso a la información por diferentes usuarios .
- Incremento en poder de cómputo.

Desventajas:

- La administración aumenta en la medida que los equipos aumentan.
- Se requiere de un lugar físico más amplio y con mejores características para su correcto desempeño.
- Se requiere más tiempo para su despliegue.

3.1.1. "**La Computación Distribuida."**[7] Es un tipo de computación en malla (Grid por sus siglas en inglés) Este tipo de computación utiliza sistemas tanto homogéneos como heterogéneos, y busca a través del procesamiento paralelo elevar la capacidad de computo de una institución.

Una de las mejores ventajas de este tipo de computación, es que se permite compartir recursos computacionales a un menor costo a través del uso de redes computacionales que le dan la capacidad de usar equipos que estén en diferentes ubicaciones geográficas.

Las Grid computacionales se han convertido recientemente en sinónimo de alto rendimiento "HPC - High Performance Computing"[8]. El objetivo principal de una malla computacional es maximizar el uso de los recursos computacionales de una

[7] COULOURIS, G. Distributed Systems: En. Concepts and Design Edition 3 (Mar 2001).P. 25.
[8] DOWD, K., C. R. SEVERANCE High performance computing. En. M. K. LOUKIDES *et al*. HPC computing. Edtion (Jun 1998). P. 26.

institución, de esta forma el aporte se mide en término de la cantidad de trabajos que puede procesar durante un periodo de tiempo.

3.1.2. "El Procesamiento Paralelo."[9] Es La habilidad de llevar múltiples operaciones o tareas de forma simultánea, uno de los principios en los que se basa la computación paralela que los problemas pueden dividirse en problemas más pequeños y ser abordados de manera simultánea.

El procesamiento paralelo también se conoce como computación paralela debido a que es necesario que existan varios procesadores físicos para hacer el procesamiento simultaneo.

El procesamiento paralelo se puede resumir como el uso simultaneo de más de una CPU o núcleo de procesador para ejecutar un programa.

Dentro del procesamiento paralelo se pueden identificar al menos tres tipos de paralelismo. El primero es paralelismo a nivel de Bits en el cual el tamaño de la palabra que maneja el procesador se incrementa desde su fabricación, este "Word size"[10] va ligado directamente a la cantidad d de bits que el procesador puede manejar, es decir 8, 16, 24, 32 o 64.

El segundo es el "paralelismo a nivel de instrucción"[11], básicamente apunta a disparar de forma síncrona varios procesos en diferentes procesadores, se debe tener en cuenta que en la actualidad los procesadores pueden lanzar amas de una instrucción a la vez.

El tercero es el paralelismo de tareas, este contiene datos paralelos en donde el mismo cálculo es realizado en diferentes set de datos, cabe notar que el paralelismo de tareas no escala con el tamaño del problema.

3.1.3 Las Redes Computacionales. Son un conjunto de equipos informáticos conectados entre sí a través de medios físicos que transportan información de diferentes destinos. Esta comunicación es posible por medio de diferentes modelos como lo es la referencia OSI. La interconexión de equipos de cómputo da origen a diferentes tipos de redes, entre ellas Internet.

[9] NAIOUF, M. Procesamiento paralelo. En. Balance dinámico de carga en algoritmos de sorting. Journal La plata (Mar 2004) P 142.
[10] WALL, D. W. Limits of instruction-level parallelism. En. ACM Journal. (Sep 1991). P 231.
[11] Ibid.

3.2 MARCO TEORICO

"Los modelos computacionales"[12] exploran posibles soluciones a una serie de problemas existentes, estos problemas son de diferente orden y en algunos casos se ven limitados por los recursos que se tengan. El proceso de modelamiento está ligado fuertemente a la solución que se pretende aplicar y de esta forma se aplican diferentes técnicas computacionales que apuntan resolver el problema a través de un análisis de datos.

Un modelo computacional al ser ejecutado en hardware, tendrá las ventajas y limitaciones del equipo con el que se cuente. Como ventajas se puede destacar el tipo de lenguaje a usar. Dependiendo de las características de hardware se tendrán también ciertas ventajas y limitaciones. So un procesador es de 32 Bits o de 64 Bits etc.

En el caso de la renderización una forma de solucionar los problemas de tiempo es dividir la tarea para vencer frente al tiempo. El uso de tecnologías que distribuyen las cargas de procesos apunta a reducir cada vez más las mismas tareas que tomaban horas.

Tabla 1. Visualización marco teórico

Marco Teórico	Tema	Implementación De Un Laboratorio Para La Realización De Pruebas De Software Distribuido Y Procesamiento En Paralelo
	Problema	Tiempo de renderizado en blender es elevado
	Hipótesis	Es posible maximizar el mismo hardware para las mismas tareas sin cambios dramáticos de hardware
	objetivo	Reducir los tiempos de renderizado explotando mejor los mismos recursos de hardware
	Área de estudio	Universidad con facultad de ingeniería de sistemas
	Unidades De Estudio	Clases de Computación distribuida y/o computación grafica

Fuente: El autor

[12] VALIANT, L. G. A bridging model for parallel computation. En. ACM Journal. (Oct 1990) P 33.

4. METODOLOGIA PROPUESTA

Es necesario dividir en secciones la metodología que se debe usar para seguir esta investigación. Es fundamental que exista una estructuración de pasos a seguir con el fin de evitar inconvenientes futuros debido a la cantidad de elementos que van a interactuar.

La etapa metodológica inicial pertenece a la categoría del analisis, como resumen es indispensable tener las siguientes ideas:

- Entender la infraestructura de la institución u organización.
- Entender el problema que se quiere abordar al pretender reducir los tiempos de computación en ejecución de tareas.
- Preparar como seria la configuración del scenario.

La segunda etapa metodológica consiste en el despliegue tecnológico de la virtualización por nucleos. Una vez configurado el escenario descrito en la guía de computación distribuida y procesamiento en paralelo (Anexo A), será posible iniciar la recopilación de datos que permitirá al usuario iniciar la etapa de experimentación y análisis.

Una vez despleagdo los elementos tecnológicos se deben iniciar.

La tercera etapa metodologica es la experimentación. El primer conjunto de experimentos se lleva a cabo de manera local teniendo en cuenta las características del host físico en el que se realizan los experimentos y asi medir el tiempo de procesamiento en un host.

El segundo set de experimentos se deben llevar a cabo usando tantos equipos como se dispongan en el laboratorio para el experimento, de esta forma se observará el impacto total que tiene la cantidad de nodos sobre el tiempo de la tarea de renderizado 3D. Resumiendo en puntos clave se muestra que se debe:

- Realizar la tarea de renderizado 3D en un solo nodo para obtener el tiempo que le toma a una unidad de cómputo, teniendo en cuenta que mas adelante el experimento tendrá clones de este host.
- Distribución de la misma tarea en diferentes nodos de manera incremental, es decir el experimento se debe llevar a cabo en 1, 2, 8, nodos con el fin de entender la tendencia de optimización frente al tiempo.

- Tabular datos de la mayor cantidad de experimentos cantidad de experimentos posibles.

La cuarta etapa metodológica requiere de la tabulación de datos que le permitirá al usuario de la guía de computación distribuida y procesamiento en paralelo, observar mediante la graficacion de datos, cual es la tendencia de tiempo que se obtiene, y la forma en la que esta paralelizacion benéfica las tareas de investigación.

La quinta etapa metodológica consiste en el análisis de los datos obtenidos en los experimentos realizados. Para este análisis será necesario crear graficas en Excel o en cualquier otro software de hojas de cálculo que permita graficar datos. Es necesario obtener tantos datos como sea posible para ver como entender:

- Elaborar graficas necesarias para entender los resultados.
- Confirmar si la cantidad de nodos es inversamente proporcional al tiempo de la tarea. Es decir a mayor cantidad de nodos, menor el tiempo de renderizado.

Las cinco etapas metodológicas de la guía de computación distribuida y procesamiento en paralelo, completan un ciclo de experimentación sugerido para comprender los beneficios de la paralelizacion de tareas. Esta metodología busca incentivar al usuario en obtener un nuevo paradgma para poder desarrollar algoritmos de programación paralela. Sólo Ingeniería de sistemas puede resolver problemas concernientes a poder de computo, es necesario ejemplificar el diseño y uso de sistemas paralelos haciendo uso de los onocimientos necesarios para abordar la resolución de problemas científicos (simulación, modelado, optimización, métodos numéricos, algorítmica).

La metodología usada en esta guía es aplicable a otras areas de investigación en otros segmentos de ingeniería como lo es Ingenieria civil, Inustrial o cualquier otra area que requiera análisis de grandes volúmenes de datos.

5. CONFIGURACIÓN PARA PRUEBAS DE SOFTWARE DISTRIBUIDO

Se describen de forma detallada los pasos para desplegar los experimentos, el documento también describe las configuraciones realizadas en el ambiente local y el ambiente del laboratorio universitario, por ultimo muestra cómo obtener resultados para graficar y entender el problema.

5.1 GUÍA RENDERIZACIÓN PARALELA 2014 V 4.3

este documento tiene de forma detallada los pasos para desplegar los experimentos, el documento también describe las configuraciones realizadas en el ambiente local y el ambiente del laboratorio universitario, por ultimo muestra cómo obtener resultados para graficar y entender el problema.

5.1.1. Antes De Empezar. Parte importante de este laboratorio consiste en clonar las VM, lo que se busca es optimizar un el uso del procesador y para esto es necesario desplegar en cada computador una cantidad ideal de VM por equipo. Es decir, si el equipo tiene 2 núcleos, se espera crear dos maquinas virtuales.

En Linux es posible ver la cantidad de núcleos de una CPU ejecutando “grep -c processor /proc/cpuinfo” como resultado obtendremos un numero que indica la cantidad de núcleos disponibles en el computador. En Windows 7 se puede ejecutar el comando “WMIC CPU Get DeviceID, NumberOfCores, NumberOfLogicalProcessors” este comando arrojará 2 resultados llamados NumeroDeNucleos/NumberOfCores y “Numero De Procesadores Logicos/Number Of Logical Procesors”. Para efectos de esta guía usaremos el valor de números lógicos.

Figura 1. Procesadores lógicos.

Fuente: El autor.

Las pruebas efectuadas en el laptop (HP Probook 4440s) en el cual se creó esta guía usa cuatro VM. Es necesario identificar la cantidad de procesadores a usar.

Teniendo en cuenta que cada máquina virtual consumirá 512 Mb de Memoria RAM, asegurarse que la suma total de las VM no supere la Memoria RAM disponible del equipo anfitrión.

5.2 CONVENCIONES

Figura 2. Convención comando

```
# Comando (# consola del súper usuario,  $ consola de un usuario diferente de súper usuario)
```

Fuente: El autor.

Figura 3. Convención texto archivo

```
# Contenido de un archivo
# El '#' es un comentario
```

Fuente: El autor.

5.3 REQUERIMIENTOS

La infraestructura de hardware requerida consta de equipos de cómputo que puedan ejecutar el software Virtual Box (disponible en la página del autor https://www.virtualbox.org).

5.3.1. Requisitos de Hardware. Se describe el hardware minimo y recomendado para la realización de la guía.

- Los equipos de cómputo deben tener minimo doble núcleo.
- Por cada núcleo de procesador se recomienda tener 512 Mb de Memoria RAM Libre.
- Tarjeta de red 10/100/1000.
- Switch y/o conexiones de red a 10/100/1000.
- Teclado.
- Mouse.
- Monitor.

5.3.2. Requisitos De Software. Los requisitos de sistema exponen los nodos sin importar el sistema operativo base. Para efectos de estas guías.

- Sistema operativo Linux o Windows 7.

- Virtualbox Ver 4.2 o superior (Disponible en la web del autor www. virtualbox.org.

- Filezilla FTP (Disponible en la web del autor https://filezilla-project.org/).

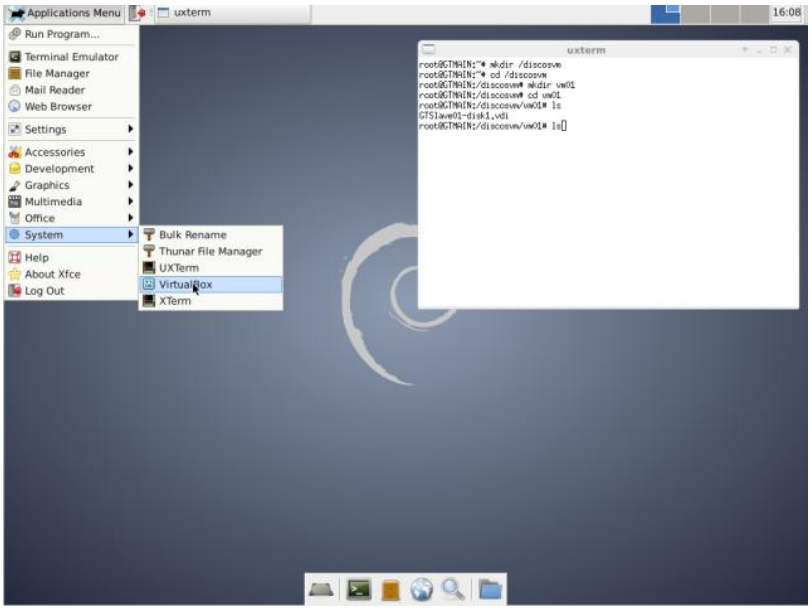
- Putty Para acceso SSH (Opcional) Disponible en la web del autor http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html)

5.4 CREACION DE NODOS VIRTUALES EN VIRTUALBOX

La VM posee todo el software necesario para paralelizar la renderización de una escena animada creada en blender. Como resultado se obtendrán los frames que representan la escena. Para iniciar es necesario copiar el archivo GTSlave01-disk1.vdi que se encuentra en la carpeta "Software Tesis\GTSlave01" en el DVD que acompaña esta guía.

5.4.1. Paso 1. Ejecutar el Icono o acceso directo de Virtualbox (Esta imagen puede ser diferente de la distribución de Linux o Windows que se esté ejecutando en el laboratorio).

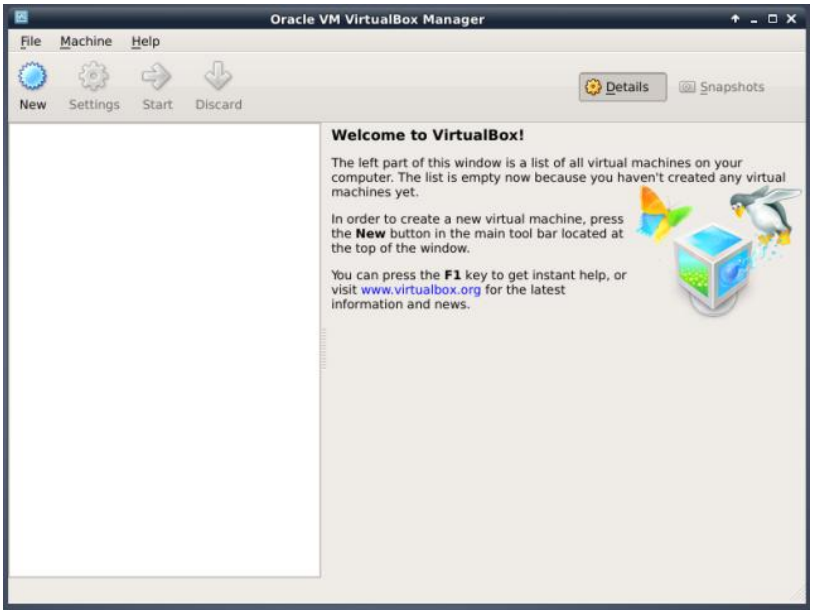
Figura 4. Acceso directo VIrtualbox



Fuente: El autor.

5.4.2. Paso 2. Una vez inicia Oracle Virtualbox, se podrá ver la ventana de bienvenida.

Figura 5. Ventana de bienvenida



Fuente: EL autor.

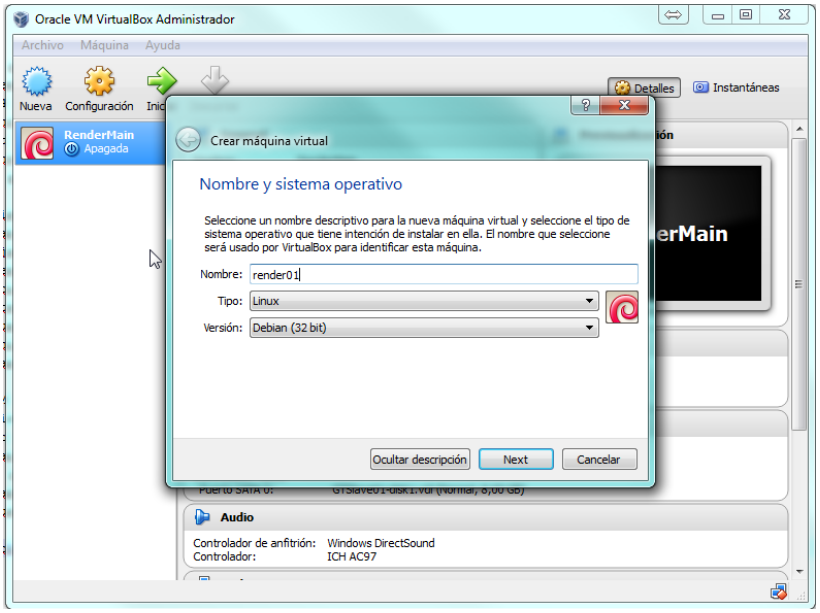
5.4.3. Paso 3. Hacer clic en New/Nueva para iniciar la creación de VM.

Figura 6. Mensaje de bienvenida.

Fuente: EL autor.

5.4.4. Paso 4. Se debe crear una VM con el nombre render0N (reemplazar N por el numero de VM que ha creado para el laboratorio) seleccione Sistema Operativo / Operación System “Linux” y en Versión escoja seleccionar Debian. Hacer clic en Next (Siguiente).

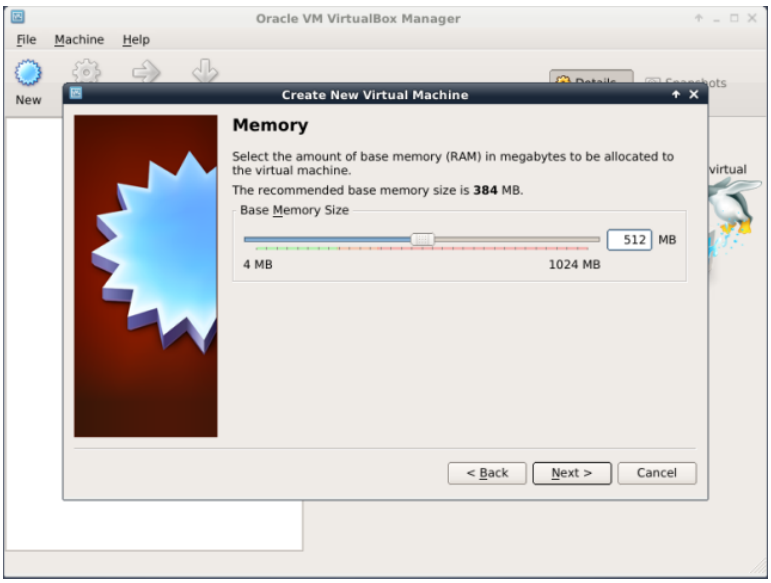
Figura 7. Creación VM.



Fuente: EL autor.

5.4.5. Paso 5. Seleccionar la cantidad de memoria de la VM. Para efectos de esta guía se configurará con un valor de 512 MB Hacer clic en next/siguiente.

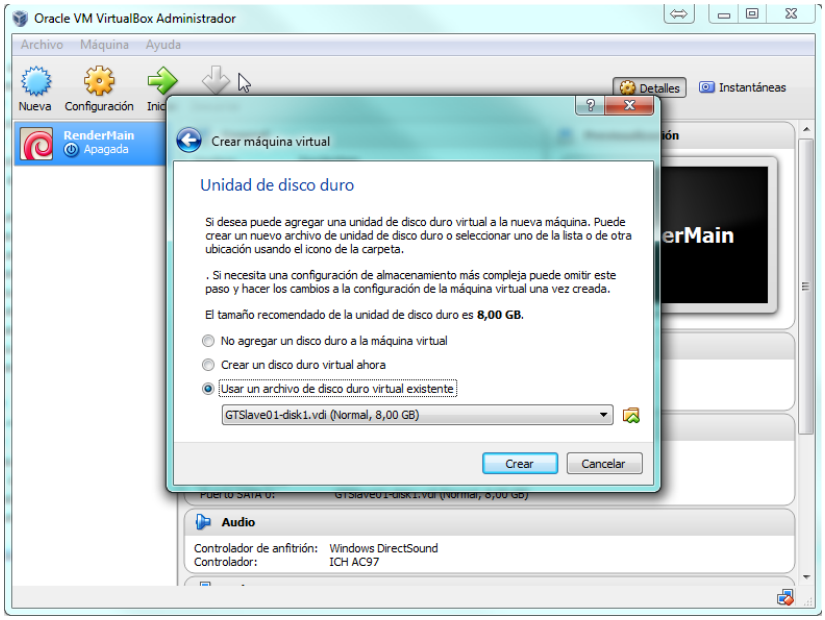
Figura 8. Cantidad memoria.



Fuente: EL autor.

5.4.6. Paso 6. En este apartado se debe seleccionar el disco duro virtual que se adjunta en esta guía.

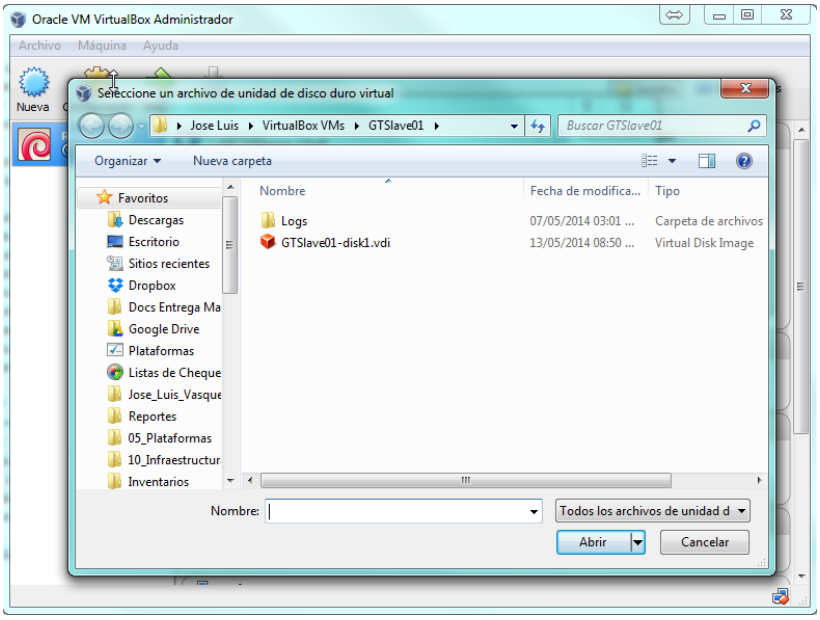
Figura 9. Selección disco duro virtual.



Fuente: EL autor.

5.4.7. Paso 7. Seleccionar el disco VDI copiado previamente. Seleccionar el archivo VDI como disco dur. Hacer clic en Open/abrir y luego en Siguiente.

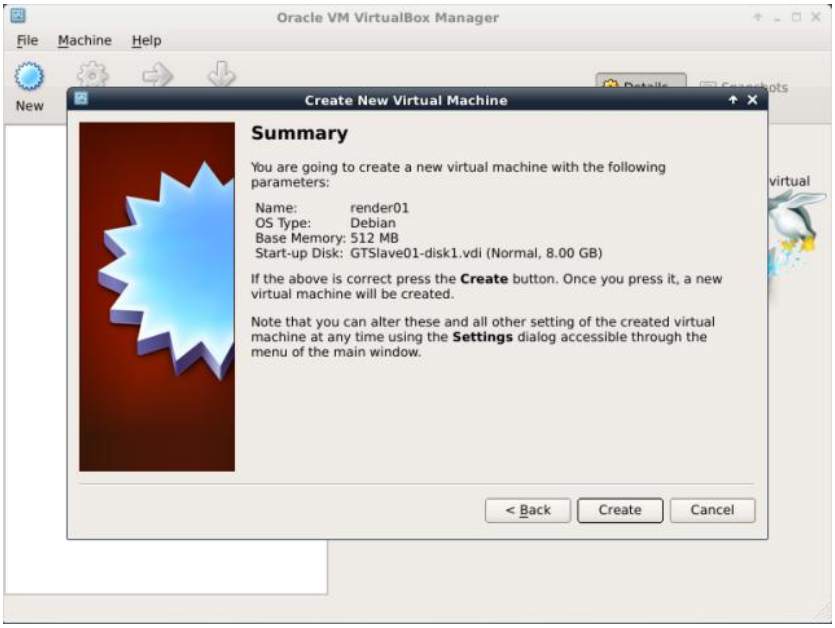
Figura 10. Uso disco virtual de laboratorio.



Fuente: EL autor.

5.4.8. Paso 8. En la ventana de summary/resumen se verán los parámetros que se han configurado para el uso de esta VM. Se debe hacer clic en Crear para que la VM quede en el registro de Virtualbox.

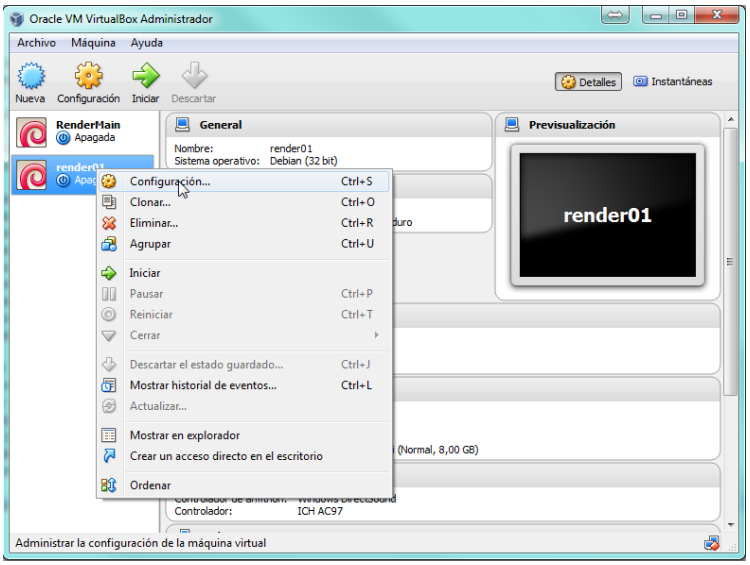
Figura 11. Resumen VM creada.



Fuente: EL autor.

5.4.9. Paso 9. Ajustar configuración de la tarjeta de red, se debe hacer clic derecho sobre la VM y hacer clic en Settings /Configuración.

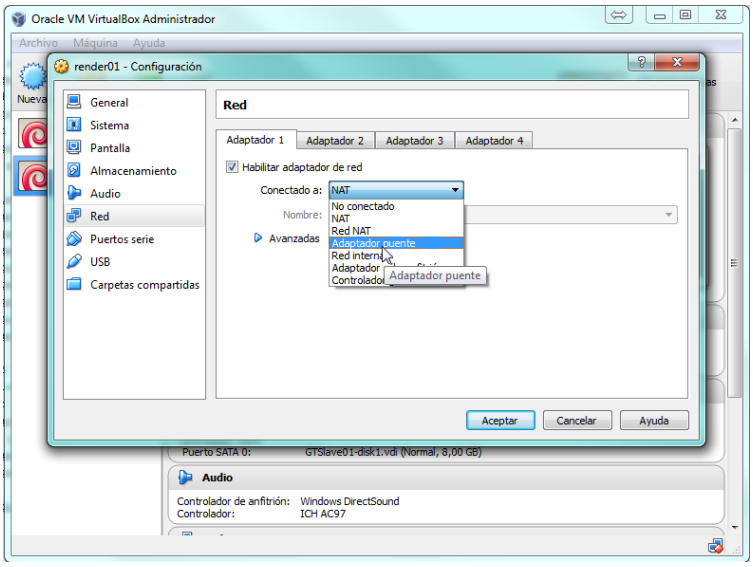
Figura 12. Propiedades VM



Fuente: EL autor.

5.4.10. Paso 10. Una vez en la ventana de Configuración /settings, hacer clic en Red/Network y en este apartado ver la pestaña Adaptador1 para configurar la tarjeta de red en modo Puente /Bridge Adapter.

Figura 13. Tipo de adaptador de red.



Fuente: EL autor.

Esto concluye la creación de una VM para desplegar el laboratorio.

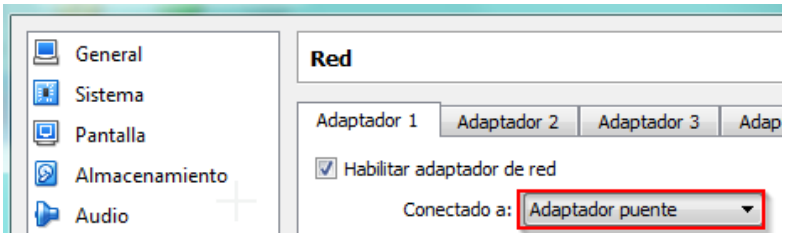
5.5 NODOS VIRTUALES UCATOLICA

La guía se encuentra desarrollada de forma genérica, sin embargo este apartado especifica la creación de nodos en la Universidad Católica teniendo en cuenta la versión de Virtualbox que está actualmente instalada en la universidad, al igual que las políticas de seguridad que esta misma posee.

El Proceso de creación de VM es el mismo, la diferencia radica en un problema detectado en los laboratorios de la universidad debido a la versión de Virtualbox. Este error hacia que las VM desplegadas en un mismo computador obtuvieran la misma IP generando un conflicto. Para resolver este problema solo basta seguir los siguientes pasos.

5.5.1. Configuración Apartado Red En Virtualbox. El apartado “Conectado a:” debe estar seleccionado como: Adaptador de puente.

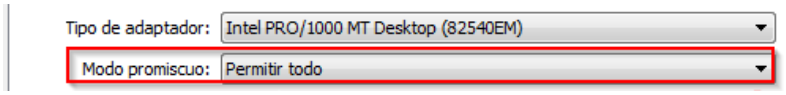
Figura 14. Adaptador de red.



Fuente: EL autor.

5.5.2. Modo Promiscuo (Virtualbox 4.3 o Superior). El apartado “Modo Promiscuo:” debe estar seleccionado como “Permitir Todo” Este apartado solo estará disponible si se ha seleccionado la opción 4.1 anteriormente mencionada.

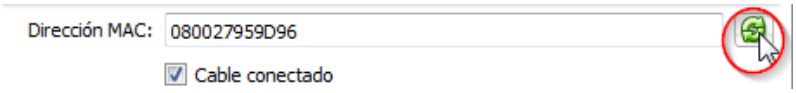
Figura 15. Modo promiscuo.



Fuente: EL autor.

5.5.3. Dirección MAC. La dirección MAC de la VM debe ser generada aleatoriamente. Se requiere presionar el botón de refrescar. Esto crea una dirección MAC nueva para evitar que dos VM tengan la misma dirección IP.

Figura 16. Reinicialización de MAC.



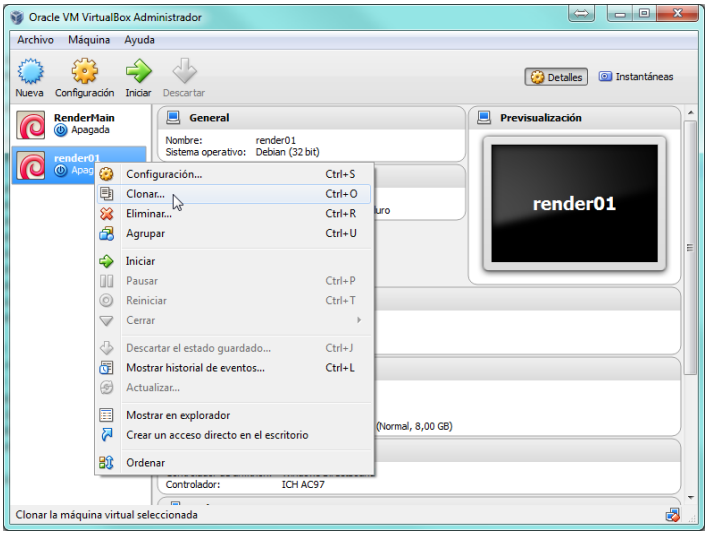
Fuente: EL autor.

5.6 CLONACIÓN DE VM

La cantidad de VM por computador obedece a la cantidad de recursos disponibles mencionados en el apartado “Antes de Empezar” descrito en esta guía.

5.6.1. Paso 1: Sobre la VM creada hacer clic derecho y sobre el menú contextual hacer clic en Clone/Clonar.

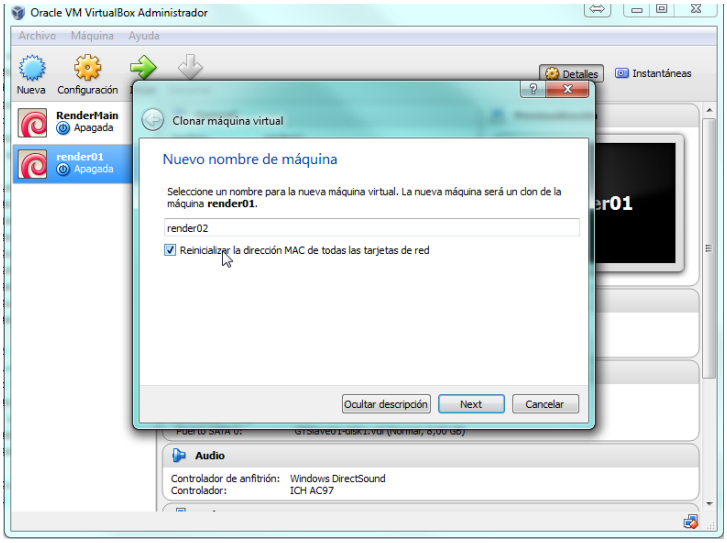
Figura 17. Propiedades VM.



Fuente: EL autor.

5.6.2. Paso 2. Siguiendo la mecánica de N mencionada anteriormente en esta guía. Marcar la opción ilustrada y hacer clic en Siguiente.

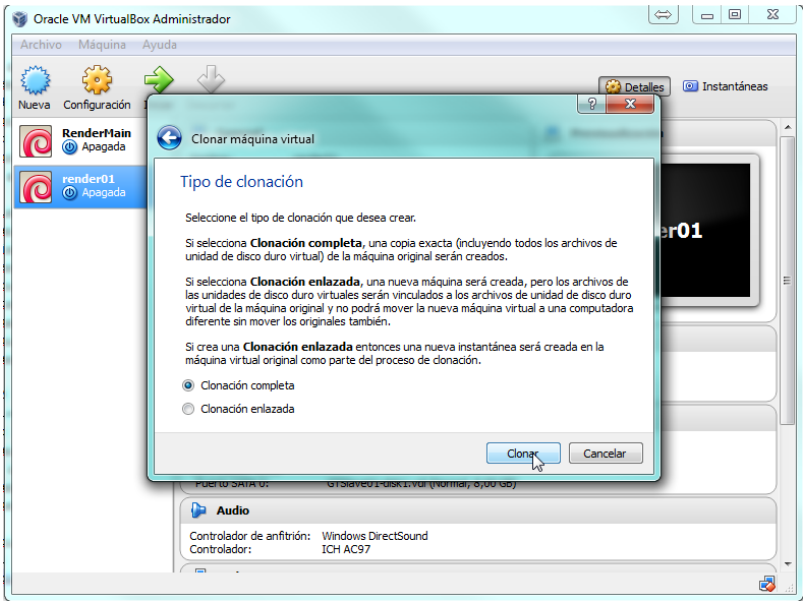
Figura 18. Nombre VM clonada.



Fuente: EL autor.

5.6.3. **Paso 3.** Seleccionar la opción Clonación Completa y hacer clic en Clonar.

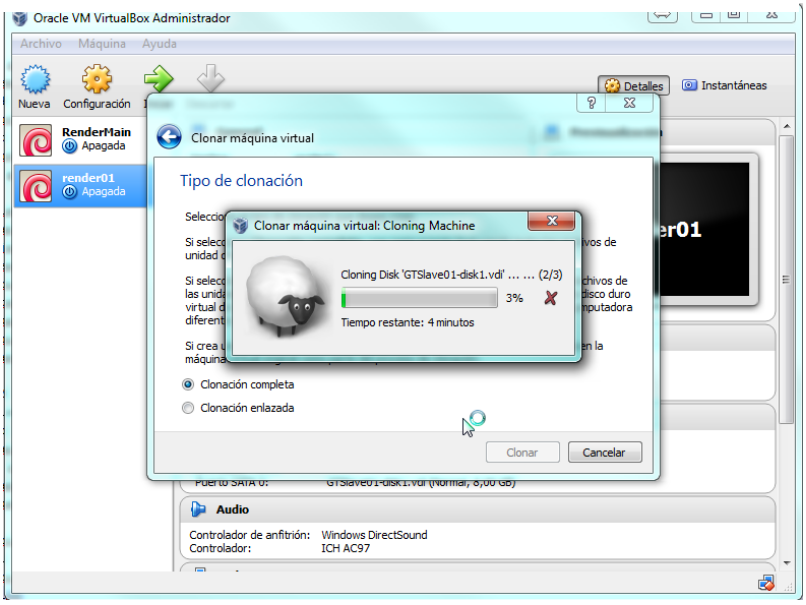
Figura 19. Inicio clonación VM.



Fuente: EL autor.

5.6.4. **Paso 4.** El asistente de clonación mostrará una barra de progreso.

Figura 20. Clonación en progreso.



Fuente: EL autor.

Una vez se tengan la cantidad de VM deseadas se debe iniciar una a una. Los datos de usuario para todas las VM son **usuario**: root **password**: ucatolica

5.7 CONFIGURACIÓN DE NODOS MULTI BLENDER

Después de iniciar las VM es necesario obtener unos datos para poder establecer comunicación entre todas las VM.

5.7.1. Obtener Dirección Ip De Cada VM. Se debe obtener datos de red de cada VM.

5.7.2. Inicio de sesión. En cada máquina y ejecutar el siguiente comando ifconfig, en rojo se observaran los datos que necesitamos capturar.

Figura 21. Salida comando ifconfig.

```
root@GTMAIN:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:d1:dc:0f
          inet addr:192.168.1.114  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fed1:dc0f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:108 errors:0 dropped:0 overruns:0 frame:0
          TX packets:60 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:13764 (13.4 KiB)  TX bytes:11302 (11.0 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Fuente: EL autor.

Se debe crear un listado de las direcciones IP que cada VM obtiene en el laboratorio.

Nota: Es posible hacer ping a otros equipos de la red para probar conectividad.

5.7.3. Establecer Comunicación Entre Las VM. Para establecer la comunicación entre VM es necesario agregar una llave pública y de esta forma establecer comunicación por SSH. Una vez se hayan desplegado las VM y se haya obtenido el listado de IPs es necesario elegir una VM para establecer la comunicación por SSH y poder hacer la tarea distribuida. La VM viene equipada

con el paquete ssh-askpass y la VM tiene una llave pública creada por efecto, esta llave pública se va a instalar en todos los nodos.

Logueado en la VM que se ha escogido será desde donde se van a lanzar los trabajos distribuidos es necesario hacer lo siguiente.

- Iniciar sesión como root.
- Ejecutar la instrucción. root@GTMAIN:~# ssh-copy-id root@192.168.1.112 teniendo en cuenta que la IP de este ejercicio será la IP de la lista que se ha obtenido.
- Responder YES cuando se pregunte si se quiere conectar.
- De pedir clave, se usa la clave que se ha establecido por defecto “ucatolica”.
- Se obtendrá el mensaje que se ha agregado la clave pública exitosamente.

Figura 22. SSH-ADD en acción.

```
Using username "root".
Linux GTSLAVE01 3.2.0-4-486 #1 Debian 3.2.54-2 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed May  7 10:10:19 2014 from oesia-hp
root@GTMAIN:~# ssh-copy-id root@192.168.1.112
The authenticity of host '192.168.1.112 (192.168.1.112)' can't be established.
ECDSA key fingerprint is ab:67:34:e4:8e:01:b7:13:af:48:34:af:d4:ce:52:42.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.112' (ECDSA) to the list of known hosts.
Now try logging into the machine, with "ssh 'root@192.168.1.112'", and check in:

  ~/.ssh/authorized_keys

to make sure we haven't added extra keys that you weren't expecting.
```

Fuente: EL autor.

Para comprobar el paso anterior se debe simplemente hacer una conexión ssh a la VM que tiene la clave publica copiada usando la instrucción “ssh 'root@192.168.1.112”.

Figura 23. Prueba ssh al nuevo nodo.

```
root@GTMAIN:~# ssh 192.168.1.112
Linux GTSLAVE01 3.2.0-4-486 #1 Debian 3.2.54-2 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed May  7 10:40:35 2014 from 192.168.1.114
root@GTSLAVE01:~#
```

Fuente: EL autor.

Se puede ver en el prompt que la VM que se está controlando ahora es la GTSLAVE01. Para salir solo se requiere escribir “exit”.

Figura 24. Salir de sesión ssh.

```
root@GTSLAVE01:~# exit
logout
Connection to 192.168.1.112 closed.
root@GTMAIN:~#
```

Fuente: EL autor.

Se debe repetir este proceso en cada VM que se quiera usar en el laboratorio. El anexo C sugiere mediante una tabla de Excel la organización de este comando para facilitar si ejecución masiva.

5.8 EJECUTANDO BLENDER DISTRIBUIDO

En la VM que se ha elegido como la VM maestra es necesario preparar multiblender para paralelizar la renderización. La VM Maestra viene equipada con todo lo necesario para hacer un ejemplo de paralelizacion al igual que como capturar los datos que se van a tomar en el laboratorio.

5.8.1. Archivo Multiblendrc. Este archivo contiene la información del cómo se va a distribuir las tareas. Este archivo tiene una estructura simple en la cual se agregan los nodos que van a ser usados en el laboratorio. Se describe y explica a

continuación la estructura del archivo multiblendrc. Este archivo de configuración se puede editar en cualquier editor como nano ejemplo: “nano ~/.multiblendrc”.

En este archivo se encuentra el apartado [main] y el apartado [node0] donde [main] hace referencia a la VM maestra y [node0] hace referencia al nodo o nodos a usar en esta guía. Si se quieren utilizar 8 VM para la tarea distribuida es necesario agregar los 8 nodos en este documento. En la siguiente imagen se explica que hace cada parte del apartado.

Figura 25. Descripción archivo multiblenderc.

```
[main] “VM Maestra”
chunks=180 “Cantidad de partes que cada nodo procesará”
nodes=1 “Cantidad de nodos que van a trabajar”
projectpath=/netrender/animacion “Ruta del archivo a procesar”
outputpath=/var/www/var/www/output “Directorio donde se van a recibir los frames procesados”
ssh=/usr/bin/ssh -Tq “Ubicacion de la function ssh”
scp=/usr/bin/scp –q “Ubicacion de la function scp”
nice=0 “Prioridad de uso de recursos, 0 es valor por defecto”

[node0] “Nombre del Nodo”
hostname=192.168.1.114 “Direccion IP del nodo”
blender=/usr/bin/blender “Ubicacion del folder donde esta blender instalado”
workdir=/netrender/animacion “Directorio de trabajo”
nice=0 0 “Prioridad de uso de recursos, 0 es valor por defecto”
ssh=/usr/bin/ssh -Tq “Ubicacion de la function ssh”
scp=/usr/bin/scp –q “Ubicacion de la function scp”
```

Fuente: EL autor.

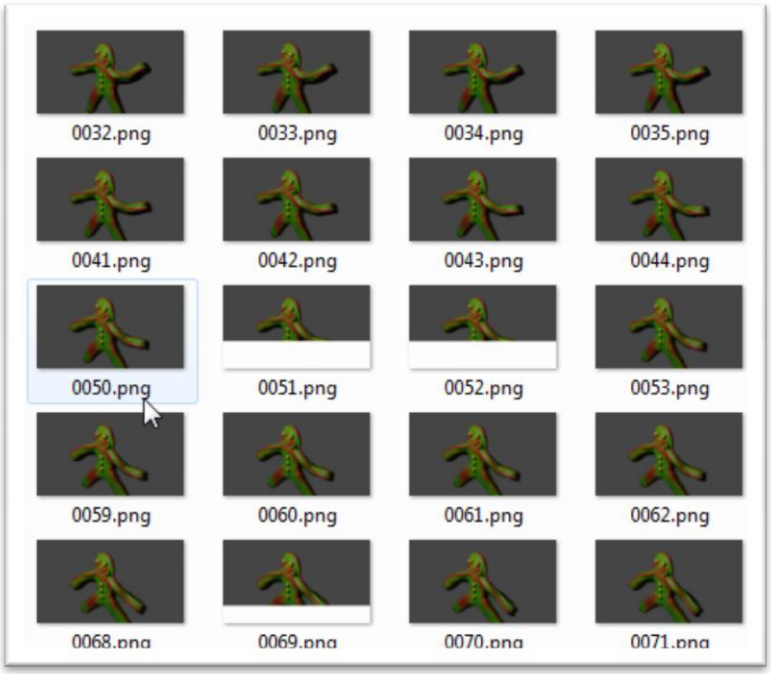
En las VM se encuentra el archivo “0015jos.blend” bajo el folder “”/netrender” correspondiente a una animación de 180 Frames de un hombre de Jengibre realizado por la clase de computación grafica. Las pruebas de esta guía se harán con esta animación. El hombre de jengibre luce como lo muestra la siguiente imagen.

Figura 26. GUS el hombre de jengibre.

Fuente: EL autor.

Al final de cada experimento es observar as imágenes renderizadas. La meta de esta investigación es acelerar el tiempo de la renderización total del proyecto, cada frame es presentada en la unidad de la VM maestra, el resultado final se puede observar en un navegador.

Figura 27. Frames GUS.



Fuente: EL autor.

5.9 Pruebas

5.9.1. Primera Prueba De 180 Frames Usando Un Solo Nodo. Para esta primera prueba solo tendremos 1 nodo el cual será la VM maestra activos en el archivo multiblendrc. El archivo de configuración multiblendrc queda configurado de la siguiente forma:

Figura 28. Configuracion multiblendrc 1 VM.

```
root@GTMAIN:~# nano  ~/.multiblendrc
```

Fuente: EL autor.

Figura 29. Configuracion multiblendrc 1 VM.

```
 [main]
chunks=180
nodes=1
projectpath=/netrender/animacion
outputpath=/var/www/var/www/output
ssh=/usr/bin/ssh -Tq
scp=/usr/bin/scp -q
nice=0

[node0]
hostname=localhost
blender=/usr/bin/blender
workdir=/netrender/animacion
nice=0
ssh=/usr/bin/ssh -Tq
scp=/usr/bin/scp -q
```

Fuente: EL autor.

De esta forma se ha configurado multiblend para que los 180 frames de la animación sean renderizadas en el nodo localhost.

Ejecutamos la instrucción “multiblend -b /netrender/0015jos.blend -s 1 -e 180” donde “multiblend” es la función de renderizado, “-b” le dice que es un archivo de blender ubicado en la ruta “/netrender/0015jos.blend”, “-s 1” es el frame en el que inicia el renderizado y “-e 180” es el frame donde termina.

Figura 30. Instrucción de renderizado.

```
root@GTMAIN:~# multiblend -b /netrender/0015jos.blend -s 1 -e 1800
```

Fuente: EL autor.

Figura 31. Salida experimento uno.

```
============================================================
Starting Multiblend 1.4
Created by Sybren A. Stüvel <sybren@stuvel.eu>
http://stuvel.eu/multiblend
============================================================

INFO MAIN: Approved [node0 localhost]
INFO localhost: rendering (1, 180)
INFO localhost: Saved /tmp/0001.png
INFO localhost: Saved /tmp/0002.png
INFO localhost: Saved /tmp/0003.png
INFO localhost: Saved /tmp/0004.png
.
.
.
INFO localhost: Saved /tmp/0175.png
INFO localhost: Saved /tmp/0176.png
INFO localhost: Saved /tmp/0177.png
INFO localhost: Saved /tmp/0178.png
INFO localhost: Saved /tmp/0179.png
INFO localhost: Saved /tmp/0180.png
INFO localhost: I'm done, no more chunks left for me.

INFO MAIN: Done multiblendering.
INFO MAIN: Start time: 2014-03-14 18:57:10.392929
INFO MAIN: End time  : 2014-03-14 19:10:21.163954
INFO MAIN: Spent     : 0:13:10.771025
```

Fuente: EL autor.

Se observa que la tarea de renderizado terminó y obtenemos los 180 frames procesados por 1 sola VM. El tiempo que toma es de 13 minutos, 10 segundos con 772 Milisegundos. Para esta prueba no se monitorea el uso de ancho de banda debido a que es una prueba local.

5.9.2. Segunda Prueba 180 Frames Usando Dos Nodos. Para la segunda prueba se han usado 2 VM (Localhost y 192.168.1.112) De la misma forma se ha configurado el archivo multiblendrc.

Figura 32 Configuración. Multiblendrc.

```
root@GTMAIN:~# nano   ~/.multiblendrc
```

Fuente: EL autor.

Figura 33. Configuración multiblendrc 2 VM.

```
[main]
chunks=90
nodes=2
projectpath=/netrender/animacion
outputpath=/var/www/var/www/output
ssh=/usr/bin/ssh -Tq
scp=/usr/bin/scp -q
nice=0

[node0]
hostname=localhost
blender=/usr/bin/blender
workdir=/netrender/animacion
nice=0
ssh=/usr/bin/ssh -Tq
scp=/usr/bin/scp -q

[node1]
hostname=192.168.1.112
blender=/usr/bin/blender
workdir=/netrender/animacion
nice=0
ssh=/usr/bin/ssh -Tq
scp=/usr/bin/scp -q
```

Fuente: EL autor.

En este caso el valor de 180 chunk se ha dividido en 2 ya que serán 2 VM las que renderizaran la tarea. A diferencia de la primera prueba, en esta prueba se tiene un nodo extra bajo 192.168.1.112.

Figura 34. Instrucción renderizado.

```
root@GTMAIN:~# multiblend -b /netrender/0015jos.blend -s 1 -e 180
```

Fuente: EL autor.

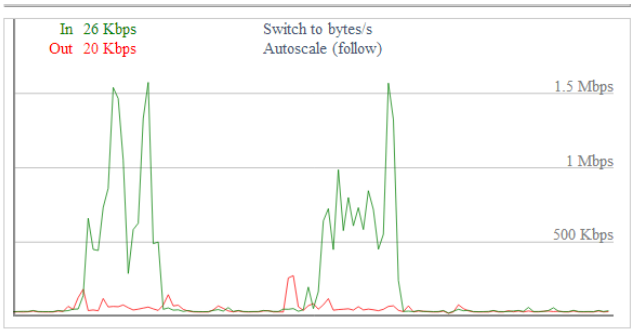
Figura 35. Salida experimento 2.

```
============================================================
Starting Multiblend 1.4
Created by Sybren A. Stüvel <sybren@stuvel.eu>
http://stuvel.eu/multiblend
============================================================

INFO MAIN: Approved [node0 localhost]
INFO MAIN: Approved [node1 192.168.1.112]
INFO localhost: rendering (1, 90)
INFO 192.168.1.112: rendering (91, 180)
INFO localhost: Saved /tmp/0001.png
INFO localhost: Saved /tmp/0002.png
INFO 192.168.1.112: Saved /tmp/0178.png
INFO localhost: Saved /tmp/0089.png
INFO 192.168.1.112: Saved /tmp/0179.png
INFO localhost: Saved /tmp/0090.png
INFO localhost: I'm done, no more chunks left for me.
INFO 192.168.1.112: Saved /tmp/0180.png
INFO 192.168.1.112: I'm done, no more chunks left for me.

INFO MAIN: Done multiblendering.
INFO MAIN: Start time: 2014-03-14 19:11:06.489596
INFO MAIN: End time  : 2014-03-14 19:17:33.687683
INFO MAIN: Spent     : 0:06:27.198087
```

Fuente: EL autor.

Al terminar la tarea de renderizado se obtiene un tiempo inferior al tiempo inicial. En este caso es de 6 minutos, 27 segundos y 199 milisegundos. En este escenario se observo el ancho de banda en la red local y se ven picos de las tareas que usaban la red.

Figura 36. Uso de red segunda prueba.



Fuente: EL autor.

5.9.3. Tercera Prueba 180 Frames Usando Tres Nodos. Prueba de multiblend.

Figura 37. Configuración multiblendrc.

```
root@GTMAIN:~# nano  ~/.multiblendrc
```

Fuente: EL autor.

Figura 38 .Configuracion multiblendrc 3 VM.

```
[main]
chunks=60
nodes=3
projectpath=/netrender/animacion
outputpath=/var/www/var/www/output
ssh=/usr/bin/ssh -Tq
scp=/usr/bin/scp -q
nice=0

[node0]
hostname=localhost
blender=/usr/bin/blender
workdir=/netrender/animacion
nice=0
ssh=/usr/bin/ssh -Tq
scp=/usr/bin/scp -q

[node1]
hostname=192.168.1.112
blender=/usr/bin/blender
workdir=/netrender/animacion
nice=0
ssh=/usr/bin/ssh -Tq
scp=/usr/bin/scp -q

[node2]
hostname=192.168.1.133
blender=/usr/bin/blender
workdir=/netrender/animacion
nice=0
ssh=/usr/bin/ssh -Tq
scp=/usr/bin/scp -q
```

Fuente: EL autor.

Figura 39. Instrucción renderizado.

```
root@GTMAIN:~# multiblend -b /netrender/0015jos.blend -s 1 -e 180
```

Fuente: EL autor.

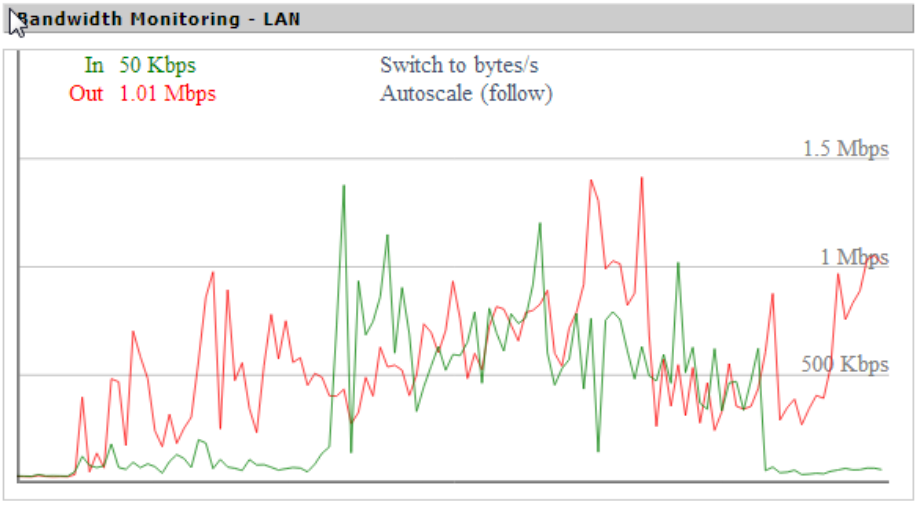
Figura 40. Salida experimento 3.

```
============================================================
Starting Multiblend 1.4
Created by Sybren A. Stüvel <sybren@stuvel.eu>
http://stuvel.eu/multiblend
============================================================

INFO MAIN: Approved [node0 localhost]
INFO MAIN: Approved [node1 192.168.1.112]
INFO MAIN: Approved [node2 192.168.1.133]
INFO localhost: rendering (1, 60)
INFO 192.168.1.112: rendering (61, 120)
INFO 192.168.1.133: rendering (121, 180)
INFO 192.168.1.112: Saved /tmp/0061.png
INFO localhost: Saved /tmp/0001.png
INFO 192.168.1.133: Saved /tmp/0121.png
INFO 192.168.1.112: Saved /tmp/0062.png
INFO localhost: Saved /tmp/0003.png
INFO 192.168.1.133: Saved /tmp/0123.png
INFO 192.168.1.112: Saved /tmp/0064.png
.
.
.
INFO localhost: Saved /tmp/0059.png
INFO 192.168.1.133: Saved /tmp/0179.png
INFO 192.168.1.112: Saved /tmp/0120.png
INFO 192.168.1.112: I'm done, no more chunks left for me.
INFO localhost: Saved /tmp/0060.png
INFO localhost: I'm done, no more chunks left for me.
INFO 192.168.1.133: Saved /tmp/0180.png
INFO 192.168.1.133: I'm done, no more chunks left for me.
INFO MAIN: Done multiblendering.
INFO MAIN: Start time: 2014-03-14 19:19:08.455391
INFO MAIN: End time  : 2014-03-14 19:23:36.739365
INFO MAIN: Spent     : 0:04:28.283974
```

Fuente: EL autor.

El tiempo obtenido fue de 4 minutos, 28 segundos y 284 milisegundos. Es posible observar que el uso de red se incremento en comparación con el segundo experimento.

Figura 41. Uso de red prueba 3.



Fuente: EL autor.

5.9.4. Cuarta Prueba 180 Frames Usando Cuatro Nodos. La cuarta prueba local usa 4 VM en diferentes tipos de virtualización como lo es KVM, es posible epecificar a cada maquina virtual que nucleo usar.

Teniendo en cuenta que el objetivo de esta guía es académico, recae sobre el sistema operativo el comportamiento de la aplicación Virtualbox. Se propone en este documento para versiones futuras, el uso de entornos de virtualización más robustos y libres como el esquema anteriormente mencionado KVM.

Los cuatro experimentos locales llevados a cabo usan computación paralela a través de la virtualización, es evidente al observar la reducción del tiempo frente a la ejecución de la misma tarea en el mismo cluster local.

Figura 42. Configuración multiblendrc.

```
root@GTMAIN:~# nano ~/.multiblendrc
```

Fuente: EL autor.

La configuración descrita a continuación hace parte del archivo mutliblendrc con 4 nodos, es importante aclarar que en el anexo 3 se encuentran tablas donde se sugiere una forma fácil para configurar estos archivos en Excel y luego pegar en el editor preferido.

Figura 43. Configuracion multiblendrc 4 VM.

```
[main]
chunks=45 “se divide el total de frames por la cantidad de nodos”
nodes=4 “Se usaran 4 nodos para 180 frames”
projectpath=/netrender/animacion
outputpath=/var/www/output
ssh=/usr/bin/ssh -Tq
scp=/usr/bin/scp -q
nice=0

[node0]
hostname=localhost
blender=/usr/bin/blender
workdir=/netrender/animacion
nice=0
ssh=/usr/bin/ssh -Tq
scp=/usr/bin/scp -q


[node1]
hostname=192.168.1.112
blender=/usr/bin/blender
workdir=/netrender/animacion
nice=0
ssh=/usr/bin/ssh -Tq
scp=/usr/bin/scp -q


[node2]
hostname=192.168.1.133
blender=/usr/bin/blender
workdir=/netrender/animacion
nice=0
ssh=/usr/bin/ssh -Tq
scp=/usr/bin/scp -q

[node3]
hostname=192.168.1.138
blender=/usr/bin/blender
workdir=/netrender/animacion
nice=0
ssh=/usr/bin/ssh -Tq
scp=/usr/bin/scp -q
```

Fuente: EL autor.

Se lanza la instrucción de renderizado.

Figura 44. Instrucción renderizado.

```
root@GTMAIN:~# multiblend -b /netrender/0015jos.blend -s 1 -e 180
```

Fuente: EL autor.

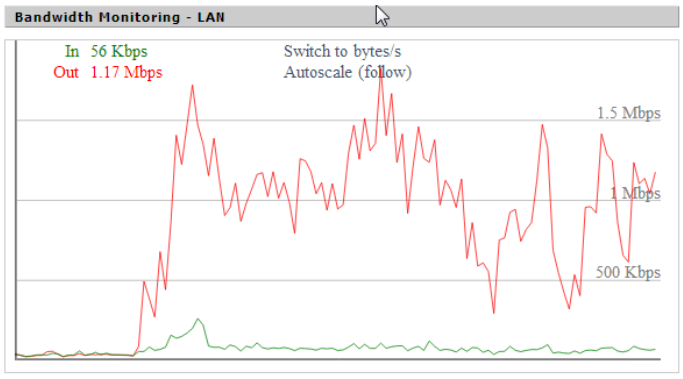
Figura 45. Salida prueba 4.

```
INFO 192.168.1.112: Saved /tmp/0046.png
INFO 192.168.1.133: Saved /tmp/0091.png
INFO localhost: Saved /tmp/0001.png
.
INFO localhost: Saved /tmp/0044.png
INFO 192.168.1.133: Saved /tmp/0135.png
INFO 192.168.1.133: I'm done, no more chunks left for me.
INFO 192.168.1.112: Saved /tmp/0090.png
INFO 192.168.1.112: I'm done, no more chunks left for me.
INFO 192.168.1.138: Saved /tmp/0180.png
INFO 192.168.1.138: I'm done, no more chunks left for me.
INFO localhost: Saved /tmp/0045.png
INFO localhost: I'm done, no more chunks left for me.
INFO MAIN: Done multiblendering.
INFO MAIN: Done multiblendering.
INFO MAIN: Start time: 2014-03-14 19:28:27.863988
INFO MAIN: End time  : 2014-03-14 19:31:59.247374
INFO MAIN: Spent     : 0:03:31.383386
```

Fuente: EL autor.

El tiempo obtenido es de 3 minutos, 31 segundos y 384 milisegundos, el consumo de red es más alto que los experimentos anteriores.

Figura 46. Uso red prueba 4.



Fuente: EL autor.

5.10 ANALIZANDO LOS DATOS

Para analizar y entender las ventajas de la computación en paralela es necesario analizar los datos obtenidos. Es importante tabular los datos usando diferente cantidad de VM.

La siguiente tabla se debe construir en el laboratorio de computación distribuida, es importante aumentar la cantidad de nodos y graficar como se muestra a continuación.

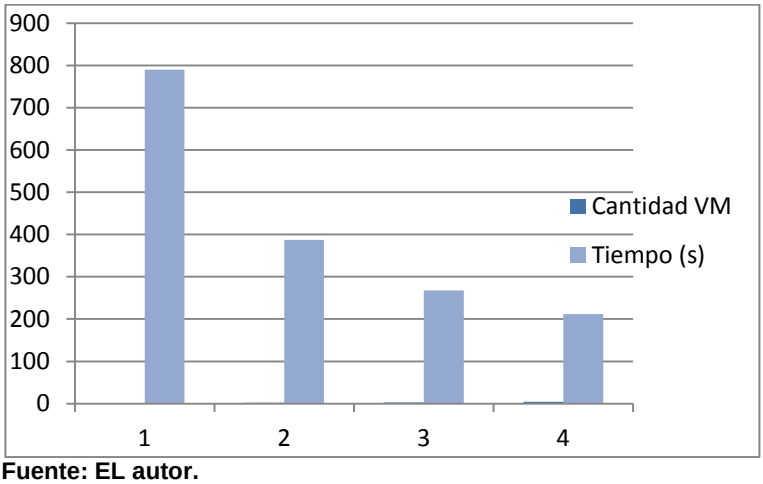
Tabla 2. Tabulación datos set de pruebas.

Maquinas Virtuales						
Frames	Procesador	Memoria	Cantidad VM	Tiempo (s)	Recursos usados	Estándar
180	Intel Core i5	512	1	790,00	49%	1%
180	Intel Core i5	512	2	387,00	49%	1%
180	Intel Core i5	512	3	268,00	49%	1%
180	Intel Core i5	512	4	212,00	49%	1%

Fuente: EL autor.

El tiempo se unifico en segundos para poder visualizar la mejora de desempeño de la tarea distribuida. Las barras de la grafica representan el tiempo total de la tarea.

Figura 47. tiempo por VM cluster local.



Fuente: EL autor.

Una vez graficado es posible ver que la misma tarea disminuye su tiempo al aumentar la cantidad de VM. En la siguiente imagen es posible observar como hay una tendencia que es inversamente proporcional a la cantidad de nodos. Es decir, a mayor cantidad de nodos menor tiempo de la tarea.

Figura 48. Tiempo set primer pruebas.

Fuente: EL autor.

Una vez se hayan analizado los datos se da final al Implementación De Un Laboratorio Para La Realización De Pruebas De Software Distribuido Y Procesamiento En Paralelo.

5.11 COMO VISUALIZAR LOS RESULTADOS

Esta guía genera las de imágenes que se almacenan en el directorio "/var/www/output", sin embargo es posible ver que las imágenes se han creado al visitar la URL de la VM maestra. Es decir en un navegador de un equipo que esté conectado a la misma subred, es posible acceder vía web y visualizar los frames resultantes.

Ejemplo**:**

Si el comando "ifconfig" muestra la IP 192.168.1.114, es posible visitar esa IP en un navegador web agregando consultaImagenes.php al final de la misma. Es decir "192.168.1.114/consultaImagenes.php". Recuerde cambiar la IP por la IP del laboratorio.

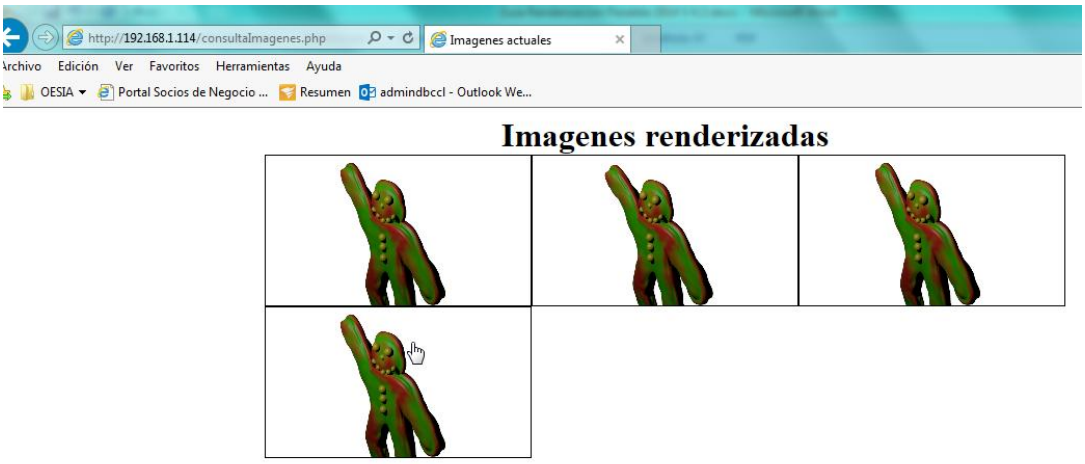
NOTA: La pagina que se muestra tiene el titulo de **Imágenes renderizadas.** Esta página tiene un control de ajax que refresca automáticamente y va mostrando las imágenes que se han renderizado.

Figura 49. Pagina local visualización imágenes.

Fuente: EL autor.

Mientras se corre el experimento se irán mostrando las imágenes renderizadas, las cuales pueden ser descargadas para edición.

Figura 50. Visualización imágenes renderizadas.



Fuente: EL autor.

6. PRUEBAS

El desarrollo de esta investigación consta de diferentes etapas, inicialmente se elabora una guía para que se implemente un laboratorio de computación paralela. La segunda etapa permite a través de la ejemplificación descrita en la guía, la toma de datos de renderización en un laboratorio de la universidad. Finalmente se tabularan los datos obtenidos para observar gráficamente como se puede optimizar una tarea común.

Toda la información sobre la guía al igual que la recolección de datos se encuentran consignadas en el Anexo A y Anexo B.

Se usa software libre[13] para esta investigación:

- GNU Debian Linux 7.4 [14] Disponible en la web del autor https://www.debian.org/.
- Virtualbox Disponible en la web del autor https://www.virtualbox.org/.
- Python2.7 Disponible en la web del autor https://www.python.org/.
- Blender 2.63 [15] http://www.blender.org/.
- Multiblender 1.14 Disponible en la web del autor http://stuvel.eu/multiblend.
- Paquete ssh-askpass[16]Disponible en cualquier distribución Linux.

6.1 ETAPA INICIAL

La virtualización emerge como una solución eficiente a problemas como los ámbitos de pruebas en los cuales las aplicaciones no podían ser probadas correctamente por problemas de licenciamiento o costos asociados con las plataformas de pruebas. El uso de esta tecnología le permite al usuario tener diferentes sistemas operativos corriendo en un mismo host sin tener que gastar dinero en equipos aislados.

---

[13] GHOSH, R. A., R. Free/libre and open source software: Survey and study. En. : GLOTT, B. KRIEGER AND G. ROBLES *et al*. Maastricht Economic Research Institute. The Netherlands Workshop 2002. P 49.

[14] MURDOCK, I. Overview of the Debian GNU/Linux system. En. Linux Journal. (Oct 1994) P 15.

[15] STÜVEL, S. A. Multiblend. [en línea] <http://stuvel.eu/multiblend> [citado en 01 de marzo de 2014].

[16] DEBIAN. How to set up ssh so you aren't asked for a password. [en línea]. <https://www.debian.org/devel/passwordlessssh.en.html> [citado en 21 de marzo de 2014].

El principio de la virtualización obedece al uso de los recursos libres que se tengan disponibles para de esta forma maximizar el uso del procesador y la memoria en el mismo equipo. Esto permite que los el mismo hardware pueda ser usado de forma más eficiente en las mismas tareas, permitiendo un crecimiento de bajo costo.

6.2 LA INVESTIGACIÓN

La presente investigación se apoya en "software libre" [17] para enseñar a través de la experimentación, el esquema de la "computación paralela"[18]. La investigación se desarrolla teniendo en cuenta:

- Entender la infraestructura de la institución.
- Entender el problema que se quiere abordar.
- Configuración del scenario.

6.3 RECOLECCIÓN DE DATOS

La recopilación de datos permitirá el propio análisis para visualizar los resultados.

- Realización de la tarea de renderizado 3D en un solo nodo para obtener el tiempo que toma por unidad de cómputo.
- Distribución de la misma tarea en diferentes nodos.

La tabulación de datos permitirá observar a través de una grafica cual es el comportamiento que se plantea obtener.

6.4 ANÁLISIS

- Tabular datos de la mayor cantidad de experimentos cantidad de experimentos posibles.
- Elaborar graficas necesarias para entender los resultados.
- Confirmar si la cantidad de nodos es inversamente proporcional al tiempo de la tarea. Es decir a mayor cantidad de nodos, menor el tiempo de renderizado.

[17] FUGGETTA, A. Open source software—an evaluation. En. Journal of Systems and Software. No 17 (Ago 2003). P 201.
[18] HWANG, K. Computer architecture and parallel processing En. Journal of the world (Feb 1984) P 20.

6.5 ESCENARIO

La investigación se lleva a cabo en el laboratorio donde existan computadores Multi-núcleo. La cantidad de nodos a usar es igual a la cantidad de núcleos que tengan los equipos. Una vez se conoce la cantidad de núcleos a usar se procede a desplegar maquinas virtuales en cada computador. Si un computador tiene 4 núcleos se deben desplegar maximo cuatro maquinas virtuales.

Al iniciar los nodos se debe establecer la comunicación sin autenticación entre los nodos, de esta forma no se requieren puertos específicos para la paralelizacion de tareas.

6.6 LOS EXPERIMENTOS

El primer experimento se lleva a cabo en un portátil de características estándares, está compuesto por un procesador Core i5 con 4 GB de RAM usando Windows 7 64 Bits como SO.

En este experimento se renderizó la misma animación de blender usando 4 nodos en un solo host. Se realizó la tarea en un nodo obteniendo un valor inicial de 790 segundos. Se incrementaron las cantidades de nodos para observar el comportamiento del mismo.

Figura 51. Tabulación datos.

HP Probook 4440s						
Frames	Proc	Mem	# VMs	T (s)	Recursos usados	Prom
180	Intel Core i5	512	1	790,00	49%	1%
180	Intel Core i5	512	2	387,00	49%	1%
180	Intel Core i5	512	3	268,00	49%	1%
180	Intel Core i5	512	4	212,00	49%	1%

Fuente: EL autor.

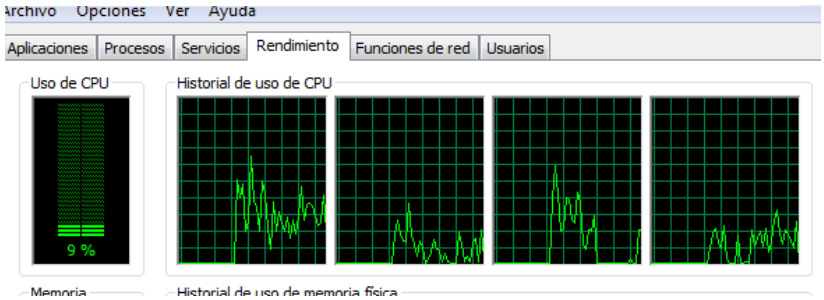
Este experimento tuvo resultados de que evidencian la optimización de tareas en la misma máquina, al bajar los tiempos de 790 segundos a 387 Segundos cuando se usaron 2 nodos y así sucesivamente hasta obtener 212 segundos con cuatro VMs.

La figura 1 explica la relación del tiempo frente a la cantidad de VMs siendo usadas. La brecha de tiempo empleada en la misma tarea comprueba la optimización de la misma.

Esta optimización obedece a el principio matemático de “proporcionalidad inversa”[19] donde las magnitudes a analizar son Cantidad de VM y Tiempo de ejecución.

Después de ejecutar la tarea en paralelo, se observó la incidencia de cantidad de VM en la reducción de tiempo sobre la misma tarea. Es visible una dramática mejora al aplicar computación paralela en el mismo host, debido a la subdivisión del uso de los núcleos del computador. Este proceso no causo un efecto negativo en el desempeño de las VM como lo muestra la figura 3.

Figura 52 .Hypervisor local.



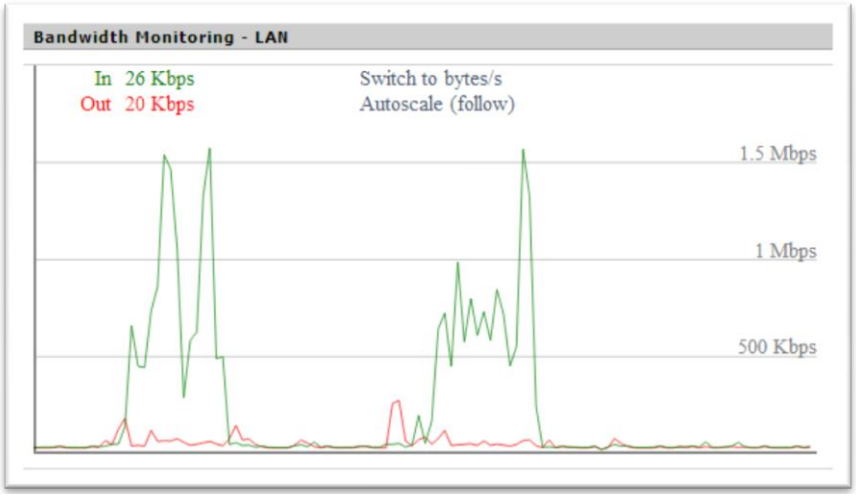
Fuente: EL autor.

Durante el primer experimento se midió el ancho de banda monitoreado. Para el primer experimento se utilizó un Roter D-Link modificado con el “firmware DD-WRT”[20] que permite observar el tipo de consumo que se está teniendo en la red ya sea LAN[21] o WAN[22]

La figura 4 muestra el consumo de la red en el primer experimento cuando se estaban usando 2 nodos, de esta forma se evidencian unos picos que corresponden a los datos que están siendo enviados y recibidos durante la ejecución de la tarea.

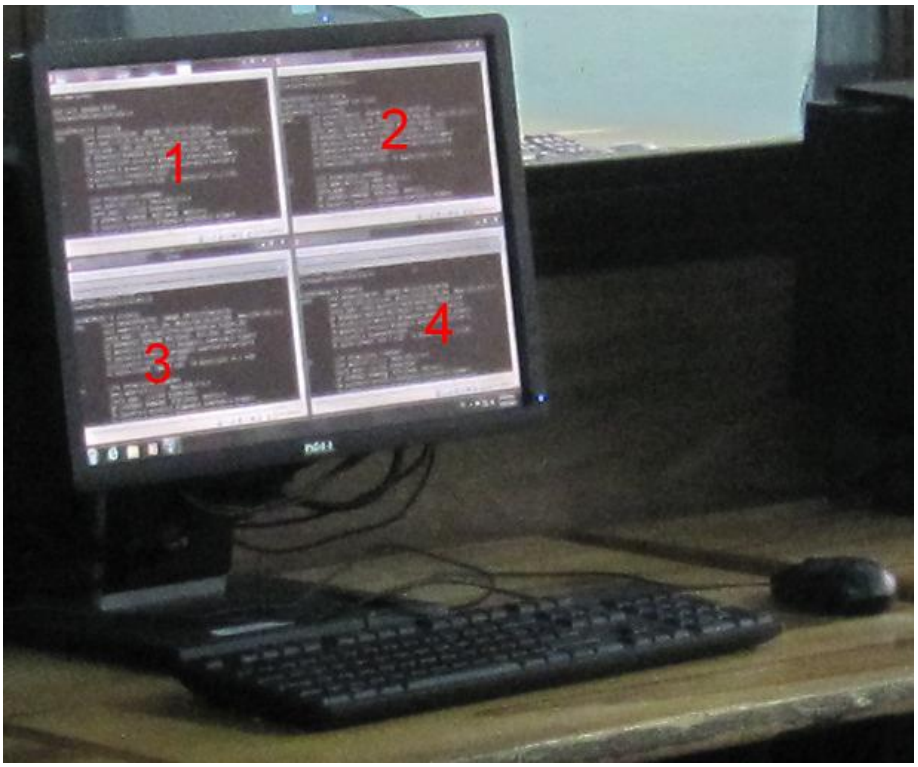
[19] WOODRUFF, Primative mathematical concepts in the chimpanzee. En. Proportionality and numerosity. (Jan 1981) P 201.
[20] BACKENS, J. A rural implementation of a 52 node mixed wireless mesh network in Macha, Zambia. En. E-Infrastructures and E-Services on Developing Countries. Springer, (Feb 2010) p. 320.
[21] ANIDO, G. J. Local area network. En. Journal Google Starbook. (Jun 1987) 221p.
[22] BALAKRISHNAN, H. M. Analyzing stability in wide-area network performance. En. ACM SIGMETRICS Performance Evaluation Review, 1997, 105 p.

Figura 53. Consumo de red.



Fuente: EL autor.

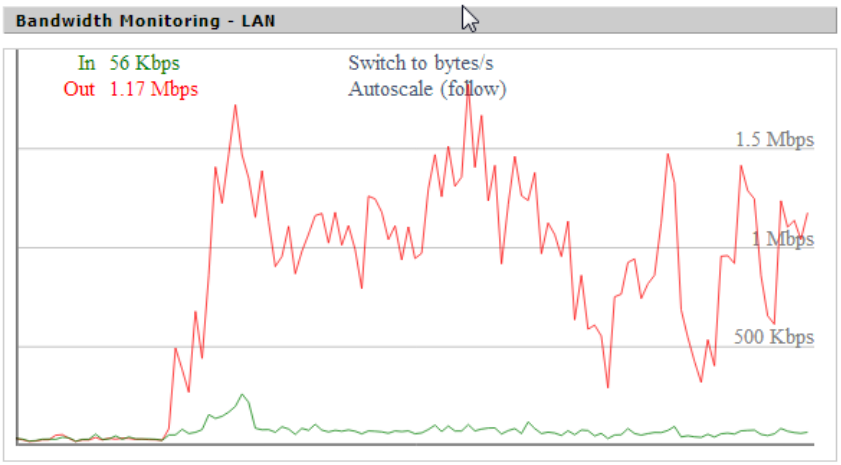
Figura 54. Clúster Local.



Fuente: EL autor.

La última prueba realizada en el primer experimento usaba cuatro VMs para renderizar la tarea inicial. Como se puede ver en la figura 5, se evidencia un incremento del ancho de banda en general mientras se está ejecutando la tarea. Esto.

Figura 55. Consumo red prueba 4.



Fuente: EL autor.

El segundo experimento llevado a cabo usó más equipos de cómputo y reafirmó los resultados obtenidos en el primer experimento. Este segundo experimento utilizo los laboratorios de la Universidad Católica de Colombia y fue posible desplegar veinte VMs en 2 salas para efectuar la misma prueba realizada en el primer experimento. El experimento se llevo a cabo usando la misma configuración de VM. "Procesador Intel Core i5"[23]. Con 512 Mb de Memoria.

Es claro hacer la observación que no se contó con hardware de red dedicado en la universidad católica de Colombia, tampoco fue posible medir los consumos de anchos de banda en las pruebas que conformaban el segundo experimento. Los resultados obtenidos muestran los beneficios de la renderización mas sin embargo queda pendiente para una futura investigación ver si comportamiento de la red en escenarios más amplios.

Tabla 3. Tabulación de datos segundo set de pruebas.

Experimento 2	
Cantidad VM	Tiempo (s)
1	790,00
2	387,00
3	268,00
4	212,00
17	75,00
18	70,00
19	65,00
20	62,00

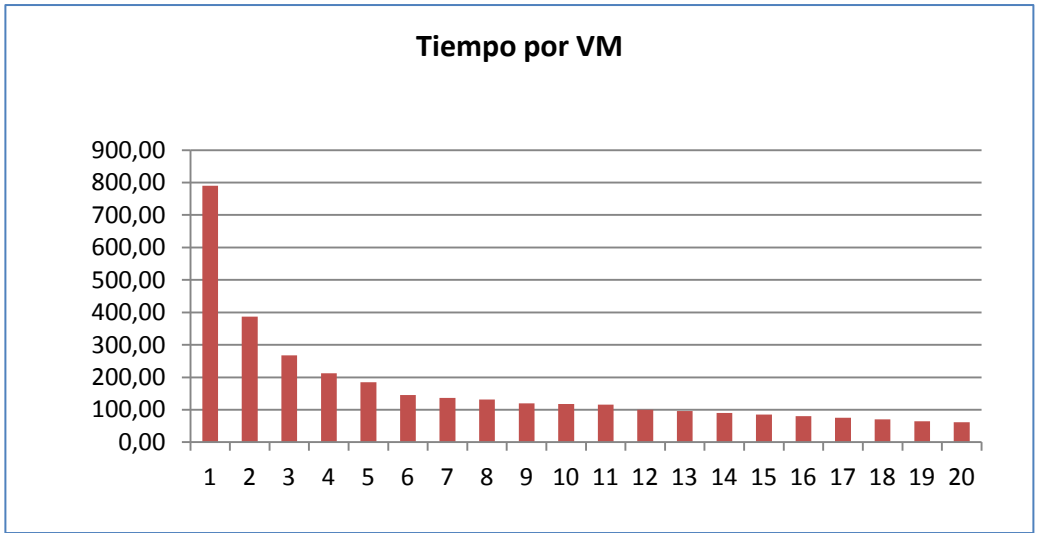
Fuente: EL autor.

[23] CASAZZA, J. Intel Core i7-800 processor series and the Intel Core i5-700 processor series based on Intel microarchitecture (Nehalem). En: White paper, Intel Corp No. 20 (Feb 2009).

Los datos de configuración de multiblend al igual que los datos tabulados de los experimentos están disponibles en el Anexo C.

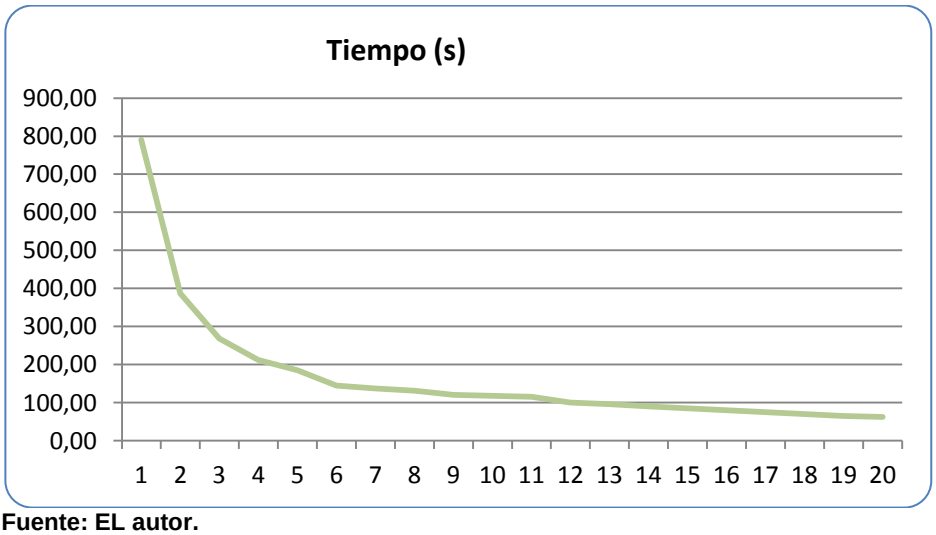
El segundo experimento comprobó los resultados esperados, en este caso se logro obtener un tiempo de 62 segundos de la tarea comprobando su reducción. Esta optimización equivale a un 92.15 % frente a la tarea ejecutada en un solo nodo. La figura 56 muestra como le tiempo empleado por VM reduce hasta un punto en el cual la reducción es minimo, mas sin embargo hay reducción de tiempo.

Figura 56. Tiempo tarea VS Cantidad VM.



Fuente: EL autor.

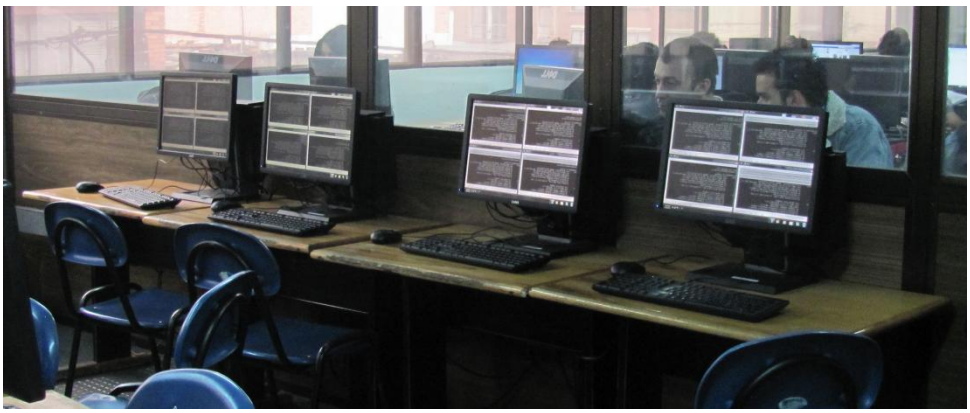
Figura 57. Tendencia tiempo.



Fuente: EL autor.

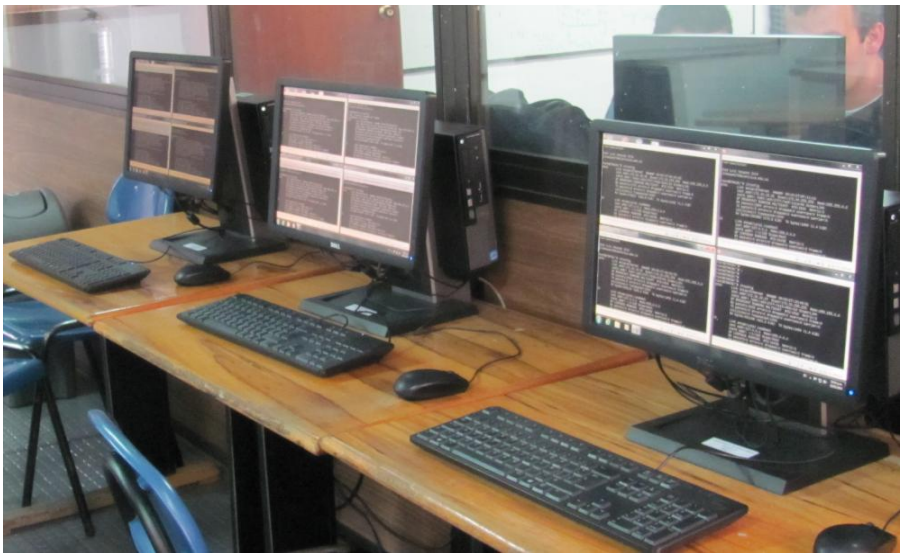
Es necesario distribuir las tareas si se quiere optimizar el hardware, la cantidad de nodos a pesar que reducen el tiempo tiene un límite de eficiencia sobre el mismo tiempo. Va a existir un punto en el cual una gran cantidad de nodos ofrezcan una ganancia de tiempo mínima. Es necesario estudiar la tarea y asignar la cantidad de nodos indicada para la misma. "Será el nivel de criticidad" [24] de la aplicación la que determine la cantidad de nodos que se pueden usar.

Figura 58. Cuatro clúster.



Fuente: EL autor.

Figura 59. Tres Cluster set de pruebas 2.



Fuente: EL autor.

[24] Duran Duran. En cómputo, r. D. S. Tecnologías de información y comunicaciones en instituciones de educación superior del sur-sureste de méxico EN. Magazine Mejico No. 01 (Dec 2005).

7. CONCLUSIONES

No fué posible validar las capacidades tecnológicas para un laboratorio de Grid en la sala de informática de la facultad de ingeniería de sistemas de la Universidad Católica de Colombia debido a temas administrativos ajenos al desarrollo de esta investigación. El anexo 2 era un elemento para validar las capacidades tecnológicas.

Fue posible implementar un servicio Grid de computación paralela en el cual se llevaron a cabo pruebas de renderizado de imágenes creadas en Blender3D, el cual arrojó resultados más que satisfactorios teniendo en cuenta que en la universidad no se contaba con una infraestructura dedicada de red. Fue posible desplegar y ver en acción el renderizado en paralelo para de esta forma entender completamente el concepto de sistemas distribuidos y los agentes que dependen del mismo.

La realización de la guía cubrió los elementos esperados de una forma fácil y simplificada, en la que no se van a requerir permisos especiales ni la instalación de software complejo optimizando el tiempo de entendimiento, ejecución y aprendizaje. La guía se desarrollo no solo pensando en la implementación sino en la forma de llevar a cabo experimentos para su tabulación y análisis.

La presente guía abre diferentes puertas para futuras investigaciones, quien siga esta guía podrá implementar un servicio Grid más robusto cambiando algunos elementos de la guía, sin afectar el concepto general de computación paralela, es decir es posible cambiar el sistema de virtualización por un sistema de Virtualización completa como VMWare[25] o si es posible llevar a cabo un sistema de paravirtualización como lo es KVM[26].

Sobre la misma guía de forma académica es posible generar algunas interfaces de configuración vía web que modifiquen el archivo multiblendrc usando php[27]. Es posible cambiar la ruta de destino por una unidad de almacenamiento externo y de esta forma optimizar el ancho de banda. Es posible crear una calculadora web para saber la cantidad de nodos que se deberían usar.

[25] ROSENBLUM, M. VMware's virtual platform™. En. Proceedings of hot chips Journal. (Oct 1999) P.185.
[26] KIVITY, A., Y. The Linux Virtual Machine Monitor. En. Proceedings of the Linux Symposium. (Mar 2007) P 225.
[27] BAKKEN, S. S. PHP Manual: Volume 2. En. iUniverse Incorporated. (Jun 2000). P 236.

Durante la realización de esta investigación solo se encontró un problema tecnológico en el laboratorio al no tener una red dedicada para poder medir el ancho de banda de los experimentos.

En el anteproyecto no se contemplaron herramientas de medición de la red para plantear diferentes esquemas de conexión y optimización para la red en la cual se desplieguen estos experimentos.

BIBLIOGRAFÍA

ANIDO, G. J. Local area network. En. Journal Google Starbook. (Jun 1987) 221p.

BACKENS, J. A rural implementation of a 52 node mixed wireless mesh network in Macha, Zambia. En: *E-Infrastructures and E-Services on Developing Countries.* Springer, (Feb 2010) p. 320.

BAKKEN, S. S. *PHP Manual: Volume 2*. En. iUniverse Incorporated. (Jun 2000). P 236.

BALAKRISHNAN, H. M. Analyzing stability in wide-area network performance. En. ACM SIGMETRICS Performance Evaluation Review, (Jun 1997). P105.

BAY, M. Transformers Revenge of the Fallen Fun Facts. [en línea]. <http://michaelbay.com/2009/06/17/transformers-revenge-of-the-fallen-fun-facts/> [citado en 21 de marzo de 2014] .

BRADNER, S. Benchmarking terminology for network interconnection devices. En: Revista Benchmarking. No. 12 (Feb., 1991).

CASAZZA, J. Intel Core i7-800 processor series and the Intel Core i5-700 processor series based on Intel microarchitecture (Nehalem). En: White paper, Intel Corp No. 20 (Feb 2009).

CORBA and dot NET remoting middleware technologies. En: International Journal of Computer Applications. No. 56 (Mar 2010).

COULOURIS, G. Distributed Systems: E:. Concepts and Design Edition 3 (Mar 2001).P. 25.

DEBIAN. How to set up ssh so you aren't asked for a password. [en línea].< https://www.debian.org/devel/passwordlessssh.en.html> [citado en 21 de marzo de 2014].

DOWD, K., C. R. SEVERANCE *High performance computing*. En: M. K. LOUKIDES et al. HPC computing. Edtion ed.: O'Reilly, 1998. p. 26.

Duran Duran. En cómputo, r. D. S. Tecnologías de información y comunicaciones en instituciones de educación superior del sur-sureste de méxico EN. Magazine Mejico No. 01 (Dec 2005).

FOSTER, I. *The Grid 2: Blueprint for a new computing infrastructure*. En. C. KESSELMAN *et al.* New Computing Infrastructure (Jan 2003) P 23-24.

FUGGETTA, A. Open source software—an evaluation. En. Revista Journal of Systems and Software. No 17 (Ago 2003).

GHOSH, R. A., R. Free/libre and open source software: Survey and study. En. : GLOTT, B. KRIEGER AND G. ROBLES et al. Maastricht Economic Research Institute on Innovation and Technology, University of Maastricht, The Netherlands, Workshop, 2002. p 49-46.

GUNION, J. F. DAWSON *The Higgs hunter's guide*. En. H. E. HABER, G. KANE AND Set al. The Guide. No. 2. (Jul 1990).

HWANG, K. Computer architecture and parallel processing En: Journal of the world (Feb 1984) P 20.

KHAN, S.Performance comparison of ICE, En. International Journal of Computer Applications HORB (Ago 2010) P 120.

KIVITY, A., Y. The Linux Virtual Machine Monitor. En. Proceedings of the Linux Symposium. (Mar 2007). P 225.

MURDOCK, I. Overview of the Debian GNU/Linux system. En. Linux Journal, (Oct 1994) P 15.

NAIOUF, M. Procesamiento paralelo. En. Balance dinámico de carga en algoritmos de sorting. Journal La plata (Mar 2004) P 142.

ROSENBLUM, M. VMware's virtual platform™. En. *Proceedings of hot chips Journal.* (Oct 1999) P.185.

STÜVEL, S. A. Multiblend. [en línea]< http://stuvel.eu/multiblend>[citado en 01 de marzo de 2014].

TESLA, N. Parallel Programming and Problem Solving with CUDA. [en línea]<https://www.youtube.com/watch?v=olvtFAArb9I>[citado en Marzo 21 de 2014].

VALIANT, L. G. A bridging model for parallel computation. En. ACM Journal. (Oct 1990) P 33.

WALL, D. W. *Limits of instruction-level parallelism*. En. ACM Journal. (Sep 1991). P 231.

WOODRUFF, Primative mathematical concepts in the chimpanzee:En. proportionality and numerosity (Jan 1981) P 201.

ANEXOS

Anexo A

Guia Renderizacion Paralela.

1 CONVENCIONES

```
# Comando (# consola del súper usuario,  $ consola de un usuario diferente de súper usuario)
```

```
# Contenido de un archivo
# El '#' es un comentario
```

2 REQUERIMIENTOS

La infraestructura de hardware requerida consta de equipos de cómputo que puedan ejecutar el software Virtual Box (disponible en la página del autor https://www.virtualbox.org).

a. ***Requisitos de Hardware.***

Se describe el hardware minimo y recomendado para la realización de la guía.

a) Los equipos de cómputo deben tener minimo doble núcleo.
b) Por cada núcleo de procesador se recomienda tener 512 Mb de Memoria RAM Libre
c) Tarjeta de red 10/100/1000
d) Switch y/o conexiones de red a 10/100/1000
e) Teclado
f) Mouse
g) Monitor

b. ***Requisitos de software***

Los requisitos de sistema exponen los nodos sin importar el sistema operativo base. Para efectos de estas guías.

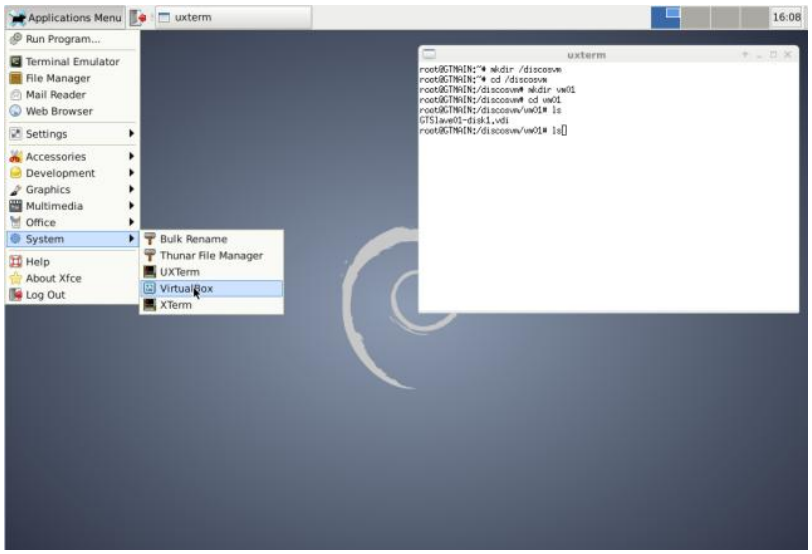
a) Sistema operativo Linux o Windows 7
b) Virtualbox Ver 4.2 o superior (Disponible en la web del autor www. virtualbox.org
c) Filezilla FTP (Disponible en la web del autor https://filezilla-project.org/)
d) Putty Para acceso SSH (Opcional) Disponible en la web del autor http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html)

3 CREACION DE NODOS VIRTUALES EN VIRTUALBOX

La VM posee todo el software necesario para paralelizar la renderización de una escena animada creada en blender. Como resultado se obtendrán los frames que representan la escena. Para iniciar es necesario copiar el archivo GTSlave01-disk1.vdi que se encuentra en la carptea “Software Tesis\GTSlave01” en el DVD que acompaña esta guía

a. ***Paso 1:***

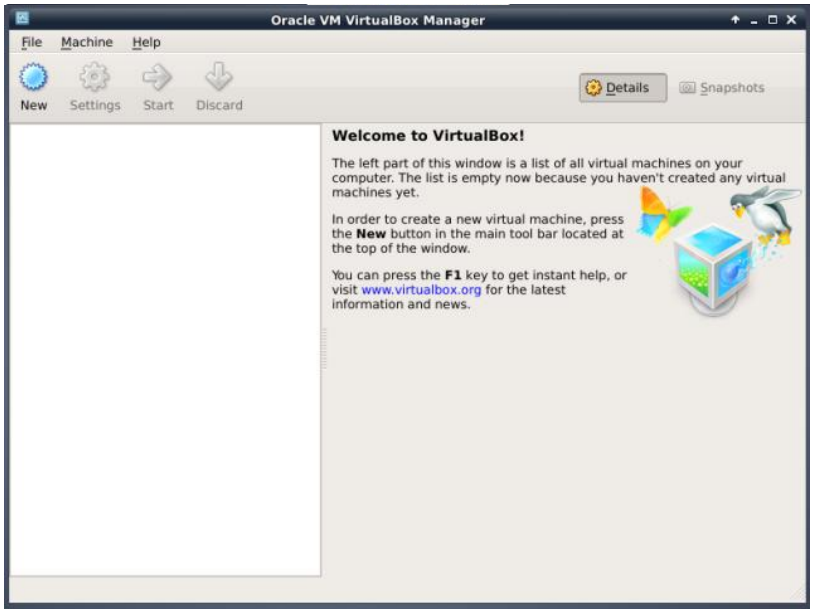
Ejecutar el Icono o acceso directo de VirtualBox (Esta imagen puede ser diferente de la distribución de Linux o Windows que se esté ejecutando en el laboratorio)



Inicio de Virtualbox XFCE Window Manager

b. ***Paso 2:***

Una vez inicie Oracle Virtualbox, se podrá ver la ventana de bienvenida



Ventana bienvenida VirtualBox

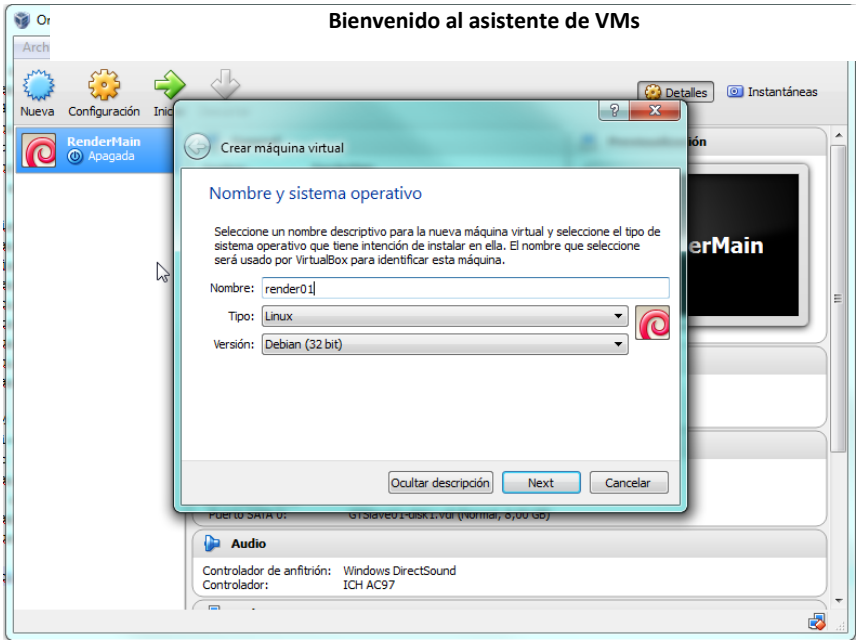
c. ***Paso 3:***

Hacer clic en New/(Nueva) para iniciar el asistente de creación de VM.

Bienvenido al asistente de VMs

d. ***Paso 4:***

Se debe crear una VM con el nombre render0N (reemplazar N por el numero de VM que ha creado para el laboratorio) y seleccione en Sistema Operativo / Operating System "Linux" y en Versión se debe seleccionar Debian. Hacer clic en Next (Siguiente)



Tipo de VM

e. ***Paso 5:***

Seleccionar la cantidad de memoria de la VM. Para efectos de esta guía se configurará con un valor de 512 Mb. Hacer clic en next/siguiente

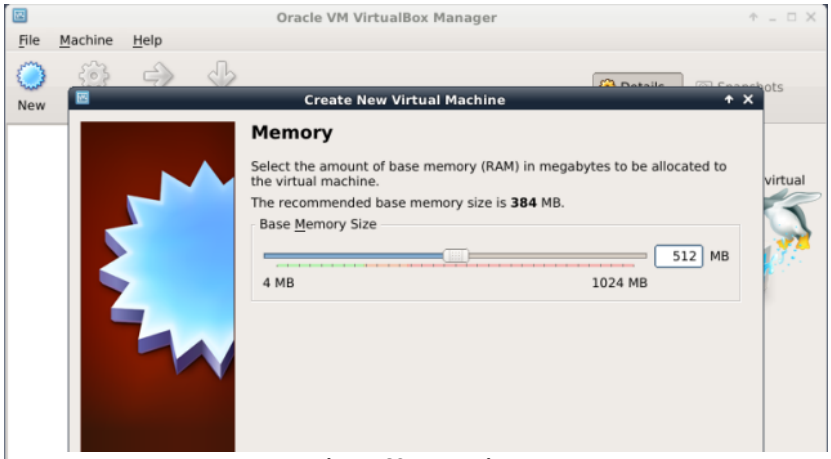
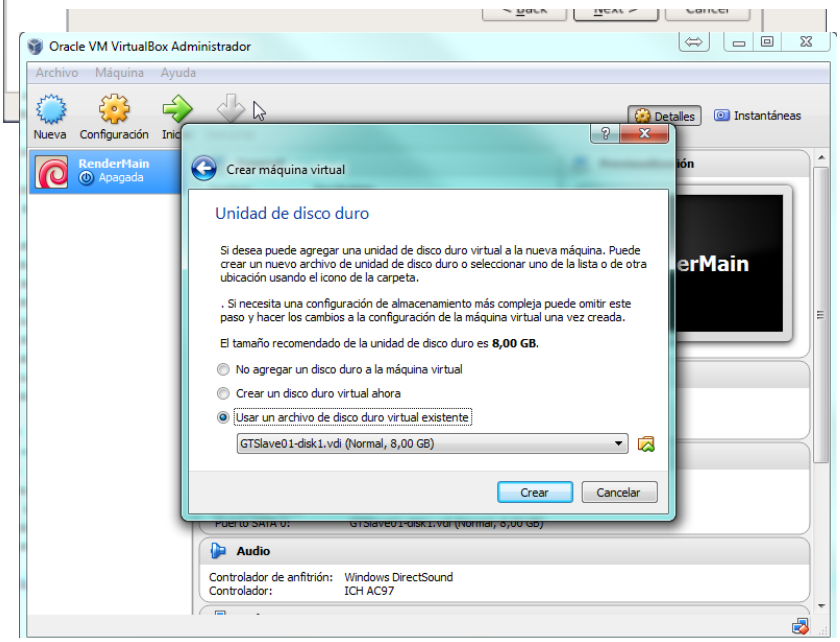


Figura 60 Memoria para VM

f. ***Paso 6:***

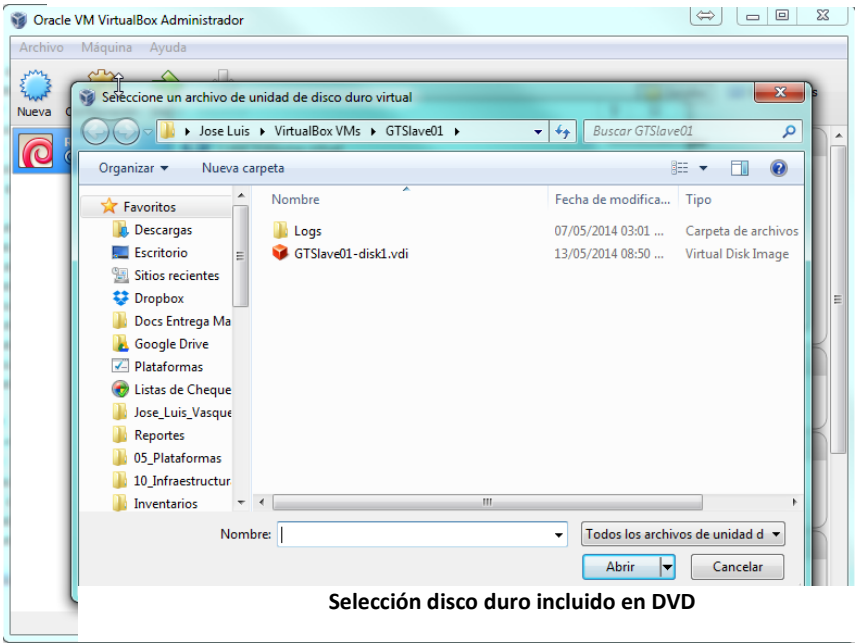
En este apartado se debe seleccionar el disco duro virtual que se adjunta en esta guía



Selección de disco virtual

g. ***Paso 7:***

Seleccionar el disco VDI que se ha copiado previamente a una ubicación del disco y seleccionar

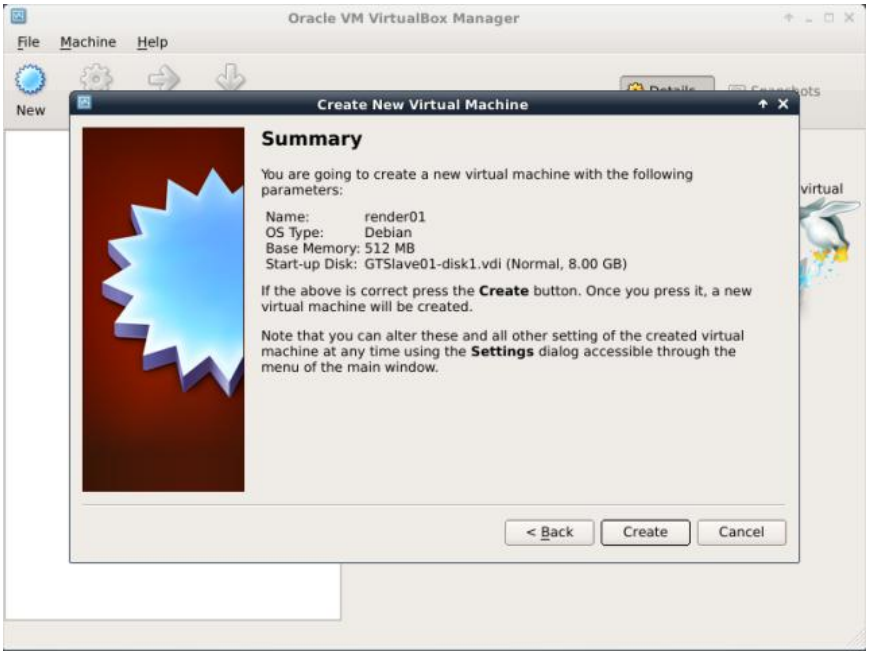


Selección disco duro incluido en DVD

el archivo VDI como disco duro de esta VM. Hacer clic en Open/abrir y luego en Siguiente

h. ***Paso 8:***

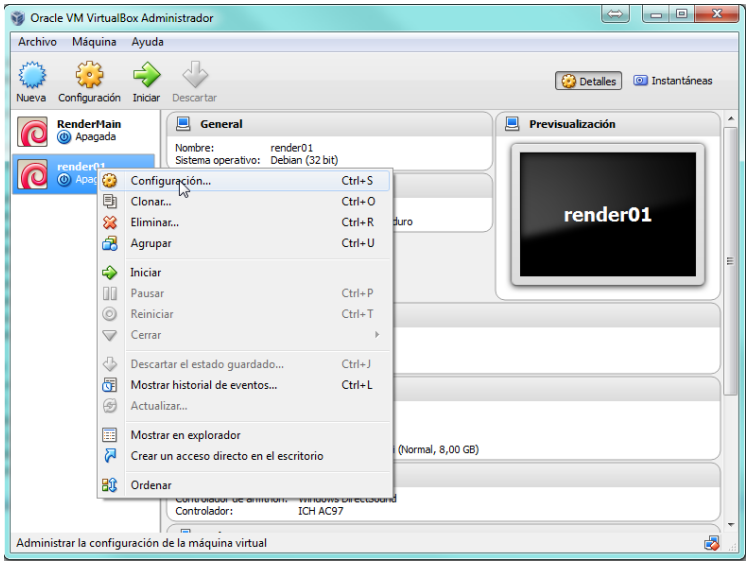
En la ventana de summary/resumen se verán los parámetros que se han configurado para el uso de esta VM. Se debe hacer clic en Crear para que la VM quede en el registro de VirtualBox



Resumen de creación

i. ***Paso 9:***

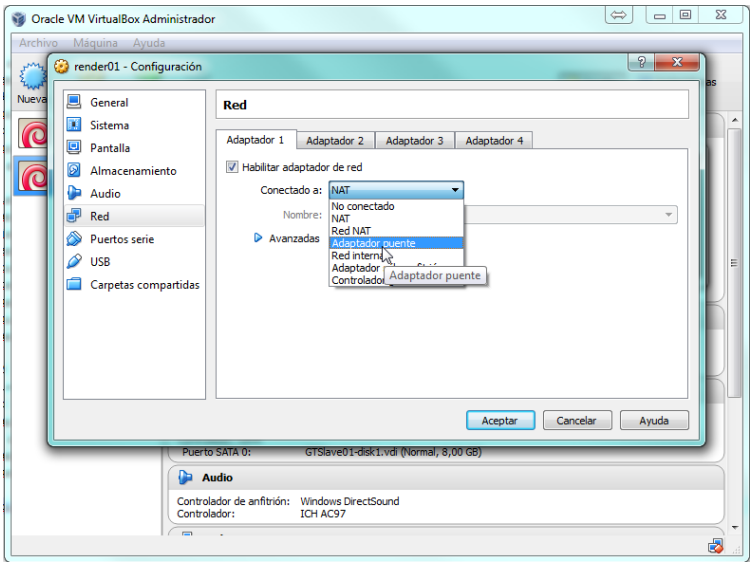
Una vez se ha creado la VM es necesario hacer un ajuste en le configuración de la tarjeta de red, se debe hacer click derecho sobre la VM recién creada y hacer clic en Settings /Configuracion



Propiedades VM creada

j. ***Paso 10:***

Una vez en la ventana de Configuración /settings, hacer clic en Red/Network y en este apartado ver la pestaña Adaptador1 para configurar la tarjeta de red en modo Puente /Bridge Adapter.



Configuración adaptador de red en modo puente

Esto concluye la creación de una VM para desplegar el laboratorio

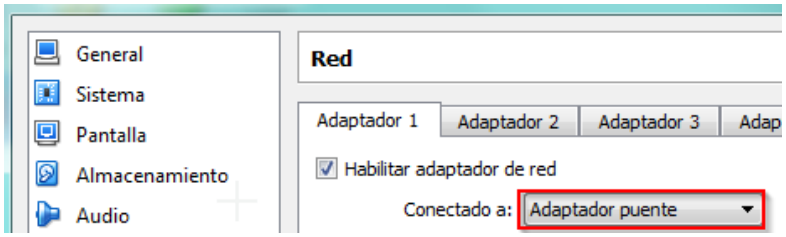
4 Nodos Virtuales UCATOLICA

La guía se encuentra desarrollada de forma genérica, sin embargo este apartado especifica la creación de nodos en la Universidad Católica teniendo en cuenta la versión de Virtualbox que está actualmente instalada en la universidad, al igual que las políticas de seguridad que esta misma posee.

El Proceso de creación de VM es el mismo, la diferencia radica en un problema detectado en los laboratorios de la universidad debido a la versión de Virtualbox. Este error hacia que las VM desplegadas en un mismo computador obtuvieran la misma IP generando un conflicto. Para resolver este problema solo basta seguir los siguientes pasos.

a. ***Configuración Apartado Red en Virtualbox***

El apartado “Conectado a:” debe estar

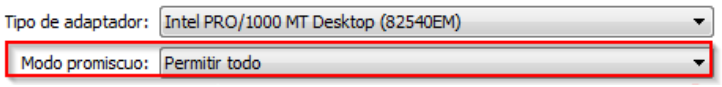


Adaptador de puente VM universidad

seleccionado como: Adaptador de puente

b. *Modo Promiscuo (Virtualbox 4.3 o superior)*

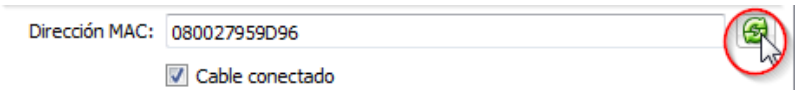
El apartado “Modo Promiscuo:” debe estar seleccionado como “Permitir Todo” Este apartado solo estará disponible si se ha seleccionado la opción 4.1 anteriormente mencionada.



Modo promiscuo

c. *Dirección MAC*

La dirección MAC de la VM debe ser generada aleatoreamente. Para esto es necesario presionar el botón de refrescar marcado en la imagen con un círculo rojo. Esto crea una dirección MAC nueva para evitar que dos VM tengan la misma dirección IP.



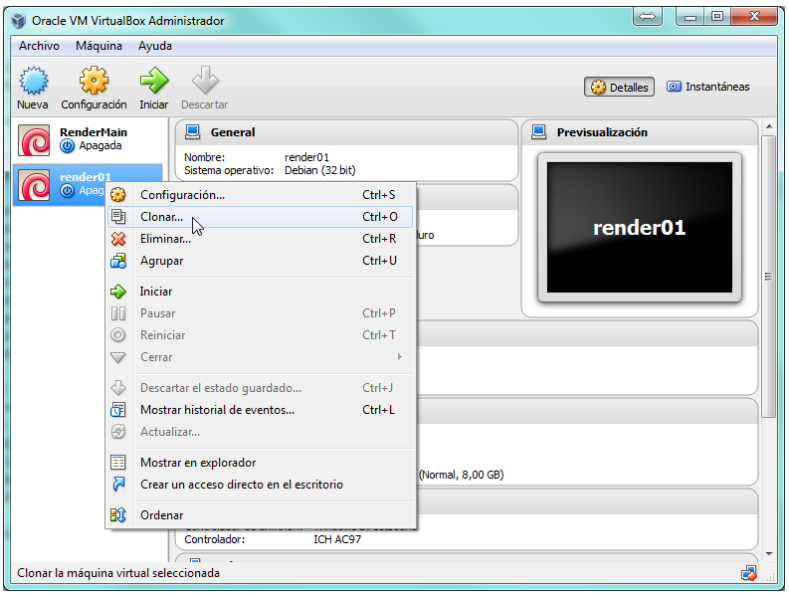
Reset de direccion MAC

5 Clonación de VM

La cantidad de VM por computador obedece a la cantidad de recursos disponibles que han sido mencionados en el apartado “Antes de Empezar” descrito en esta guía.

a. *Paso 1:*

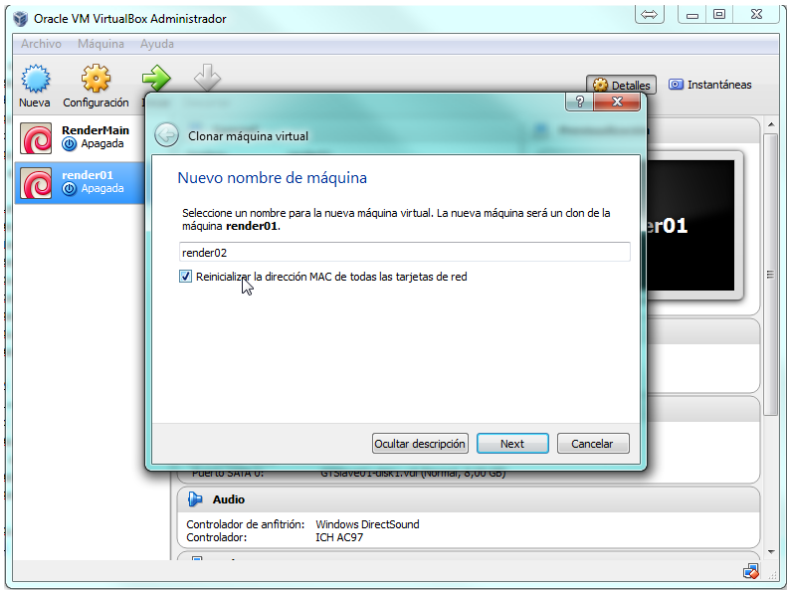
Sobre la VM creada hacer clic derecho y sobre el menú contextual hacer clic en Clone/Clonar



Menu contextual "Clonar"

b. ***Paso 2:***

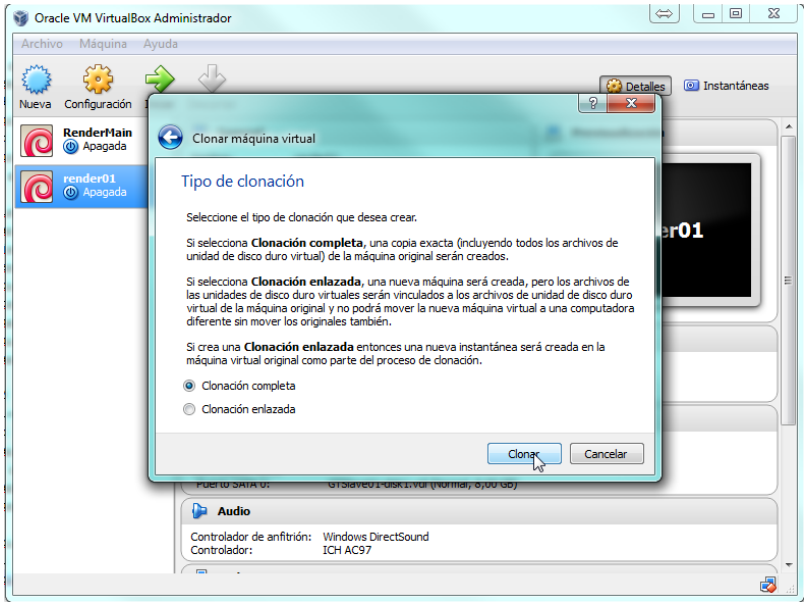
Se abrirá el asistente de clonación, se requiere nombrar la nueva VM siguiendo la mecánica de N mencionada anteriormente en esta guía. Esta VM se llamara render02. Marcar la opción de “Reinicializar la dirección MAC de todas las tarjetas de red” Posteriormente hacer clic en Siguiente.



Nombre & reinicialización de MAC

c. ***Paso 3:***

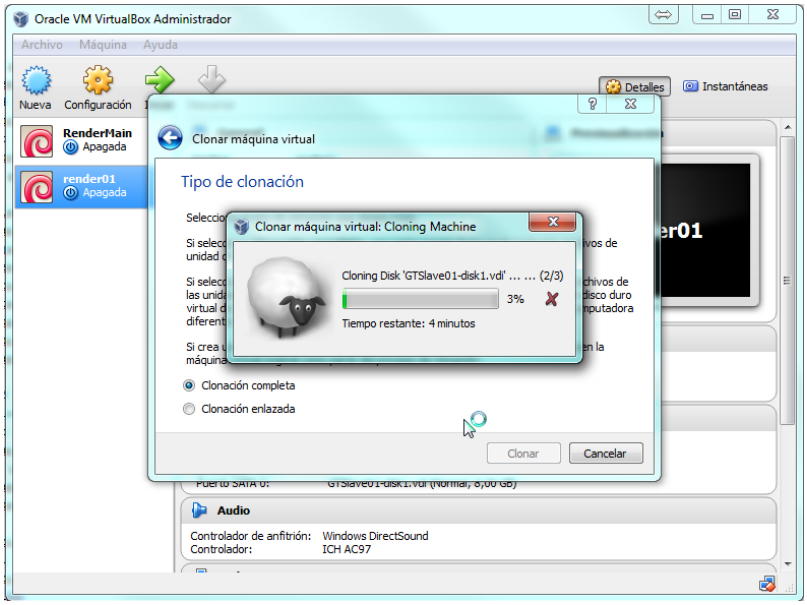
Seleccionar la opción Clonación Completa y hacemos clic en Clonar



Tipo de clonación

d. ***Paso 4:***

El asistente de clonación mostrará una barra de progreso



Clonación en proceso

Una vez se tengan la cantidad de VM deseadas se debe iniciar una a una. Los datos de usuario para todas las VM son **usuario**: root **password**: ucatolica

6 Configuración de Nodos Multi Blender

Después de iniciar las VM es necesario obtener unos datos para poder establecer comunicación entre todas las VM

a. ***Obtener Dirección IP de cada VM.***

Se debe iniciar sesión en cada máquina y ejecutar el siguiente comando ifconfig, en rojo se observaran los datos que necesitamos capturar

```
root@GTMAIN:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:d1:dc:0f
          inet addr:192.168.1.114  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fed1:dc0f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:108 errors:0 dropped:0 overruns:0 frame:0
          TX packets:60 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:13764 (13.4 KiB)  TX bytes:11302 (11.0 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Ejecución comando "ifconfig"

Se debe crear un listado de las direcciones IP que cada VM obtiene en el laboratorio.

Nota: Es posible hacer ping a otros equipos de la red para probar conectividad.

b. *Establecer comunicación entre las VM*

Para establecer la comunicación entre VM es necesario agregar una llave pública y de esta forma establecer comunicación por SSH. Una vez se hayan desplegado las VM y se haya obtenido el listado de IPs es necesario elegir una VM para establecer la comunicación por SSH y poder hacer la tarea distribuida. La VM viene equipada con el paquete ssh-askpass y la VM tiene una llave pública creada por efecto, esta llave pública se va a instalar en todos los nodos.

Logueado en la VM que se ha escogido será desde donde se van a lanzar los trabajos distribuidos es necesario hacer lo siguiente.

1. Iniciar sesión como root
2. Ejecutar la instrucción. root@GTMAIN:~# ssh-copy-id root@192.168.1.112 teniendo en cuenta que la IP de este ejercicio será la IP de la lista que se ha obtenido.
3. Responder YES cuando se pregunte si se quiere conectar

4. De pedir clave, se usa la clave que se ha establecido por defecto “ucatolica”
5. Se obtendrá el mensaje que se ha agregado la clave pública exitosamente.

```
Using username "root".
Linux GTSLAVE01 3.2.0-4-486 #1 Debian 3.2.54-2 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed May  7 10:10:19 2014 from oesia-hp
root@GTMAIN:~# ssh-copy-id root@192.168.1.112
The authenticity of host '192.168.1.112 (192.168.1.112)' can't be established.
ECDSA key fingerprint is ab:67:34:e4:8e:01:b7:13:af:48:34:af:d4:ce:52:42.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.112' (ECDSA) to the list of known hosts.
Now try logging into the machine, with "ssh 'root@192.168.1.112'", and check in:

  ~/.ssh/authorized_keys

to make sure we haven't added extra keys that you weren't expecting.
```

Salida ssh-copy-id

Para comprobar el paso anterior se debe simplemente hacer una coneccion ssh a la VM que tiene la clave publica copiada usando la instrucción “ssh 'root@192.168.1.112”

```
root@GTMAIN:~# ssh 192.168.1.112
Linux GTSLAVE01 3.2.0-4-486 #1 Debian 3.2.54-2 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed May  7 10:40:35 2014 from 192.168.1.114
root@GTSLAVE01:~#
```

Conexión ssh al nuevo nodo

Se puede ver en el prompt que la VM que se está controlando ahora es la GTSLAVE01. Para salir solo se requiere escribir “exit”

```
root@GTSLAVE01:~# exit
logout
Connection to 192.168.1.112 closed.
root@GTMAIN:~#
```

Ejecución comando "exit"

Se debe repetir este proceso en cada VM que se quiera usar en el laboratorio.

7 EJECUTANDO BLENDER DISTRIBUIDO

En la VM que se ha elegido como la VM maestra es necesario preparar multiblender para paralelizar la renderización. La VM Maestra viene equipada con todo lo necesario para hacer un ejemplo de paralelizacion al igual que como capturar los datos que se van a tomar en el laboratorio.

a. *Archivo multiblendrc*

Este archivo contiene la información del cómo se va a distribuir las tareas. Este archivo tiene una estructura simple en la cual se agregan los nodos que van a ser usados en el laboratorio. Se describe y explica a continuación la estructura del archivo multiblendrc. Este archivo de configuración se puede editar en cualquier editor como nano ejemplo: “nano ~/.multiblendrc”.
En este archivo se encuentra el apartado [main] y el apartado [node0] donde [main] hace referencia a la VM maestra y [node0] hace referencia al nodo o nodos a usar en esta guía. Si se quieren utilizar 8 VM para la tarea distribuida es necesario agregar los 8 nodos en este documento. En la siguiente imagen se explica que hace cada parte del apartado.

```
[main]  "VM Maestra"
chunks=180  "Cantidad de partes que cada nodo procesará"
nodes=1 "Cantidad de nodos que van a trabajar"
projectpath=/netrender/animacion "Ruta del archivo a procesar"
outputpath=/var/www/var/www/output  "Directorio donde se van a recibir los frames procesados"
ssh=/usr/bin/ssh -Tq "Ubicacion de la function ssh"
scp=/usr/bin/scp –q "Ubicacion de la function scp"
nice=0 "Prioridad de uso de recursos, 0 es valor por defecto"

[node0] "Nombre del Nodo"
hostname=192.168.1.114 "Direccion IP del nodo"
blender=/usr/bin/blender "Ubicacion del folder donde esta blender instalado"
workdir=/netrender/animacion "Directorio de trabajo"
nice=0 0 "Prioridad de uso de recursos, 0 es valor por defecto"
ssh=/usr/bin/ssh -Tq "Ubicacion de la function ssh"
scp=/usr/bin/scp –q "Ubicacion de la function scp"
```

Contenido archivo multiblendrc

EN la VM maestra se encuentra el archivo "0015jos.blend" bajo el folder ""/netrender" correspondiente a una animación de 180 Frames de un hombre de Jengibre realizdo por la clase de computación grafica. Las pruebas de esta guía se harán con esta animación.

8 PRUEBAS

a. *Primera prueba de 180 frames usando un solo nodo*

Para esta primera prueba solo tendremos 1 nodo el cual será la VM maestra activos en el archivo multiblendrc. El archivo de configuración multiblendrc queda configurado de la siguiente forma:

```
root@GTMAIN:~# nano  ~/.multiblendrc
```

Ejecución archivo multiblendrc

```
 [main]
chunks=180
nodes=1
projectpath=/netrender/animacion
outputpath=/var/www/var/www/output
ssh=/usr/bin/ssh -Tq
scp=/usr/bin/scp -q
nice=0

[node0]
hostname=localhost
blender=/usr/bin/blender
workdir=/netrender/animacion
nice=0
ssh=/usr/bin/ssh -Tq
scp=/usr/bin/scp -q
```

Configuración multiblendrc local

De esta forma se ha configurado multiblend para que los 180 frames de la animación sea renderizada en el nodo localhost.

Ejecutamos la instrucción “multiblend -b /netrender/0015jos.blend -s 1 -e 180” donde “multiblend” es la función de renderizado, “-b” le dice que es un archivo de blender ubicado en la ruta “/netrender/0015jos.blend”, “-s 1” es el frame en el que inicia el renderizado y “-e 180” es el frame donde termina.

```
root@GTMAIN:~# multiblend -b /netrender/0015jos.blend -s 1 -e 1800
```

Instrucción multiblend

```
============================================================
Starting Multiblend 1.4
Created by Sybren A. Stüvel <sybren@stuvel.eu>
http://stuvel.eu/multiblend
============================================================

INFO MAIN: Approved [node0 localhost]
INFO localhost: rendering (1, 180)
INFO localhost: Saved /tmp/0001.png
INFO localhost: Saved /tmp/0002.png
INFO localhost: Saved /tmp/0003.png
INFO localhost: Saved /tmp/0004.png
.
.
.
INFO localhost: Saved /tmp/0175.png
INFO localhost: Saved /tmp/0176.png
INFO localhost: Saved /tmp/0177.png
INFO localhost: Saved /tmp/0178.png
INFO localhost: Saved /tmp/0179.png
INFO localhost: Saved /tmp/0180.png
INFO localhost: I'm done, no more chunks left for me.

INFO MAIN: Done multiblendering.
INFO MAIN: Start time: 2014-03-14 18:57:10.392929
INFO MAIN: End time  : 2014-03-14 19:10:21.163954
INFO MAIN: Spent     : 0:13:10.771025
```

Salida ejecución multiblender

Se observa que la tarea de renderizado terminó y obtenemos los 180 frames procesados por 1 sola VM. El tiempo que toma es de 13 minutos, 10 segundos con 772 Milisegundos. Para esta prueba no se monitorea el uso de ancho de banda debido a que es una prueba local.

b. ***Segunda prueba 180 frames usando dos nodos***

Para la segunda prueba se han usado 2 VM (Localhost y 192.168.1.112)
De la misma forma se ha configurado el archivo multiblendrc

```
root@GTMAIN:~# nano   ~/.multiblendrc
```

Edición archivo multiblendrc

```
[main]
chunks=90
nodes=2
projectpath=/netrender/animacion
outputpath=/var/www/var/www/output
ssh=/usr/bin/ssh -Tq
scp=/usr/bin/scp -q
nice=0

[node0]
hostname=localhost
blender=/usr/bin/blender
workdir=/netrender/animacion
nice=0
ssh=/usr/bin/ssh -Tq
scp=/usr/bin/scp -q

[node1]
hostname=192.168.1.112
blender=/usr/bin/blender
workdir=/netrender/animacion
nice=0
ssh=/usr/bin/ssh -Tq
scp=/usr/bin/scp -q
```

Configuración multiblendrc 2 VM

En este caso el valor de 180 chunk se ha dividido en 2 ya que serán 2 VM las que renderizaran la tarea. A diferencia de la primera prueba, en esta prueba se tiene un nodo extra bajo 192.168.1.112

```
root@GTMAIN:~# multiblend -b /netrender/0015jos.blend -s 1 -e 180
```

Edición archivo multiblendrc

```
============================================================
Starting Multiblend 1.4
Created by Sybren A. Stüvel <sybren@stuvel.eu>
http://stuvel.eu/multiblend
============================================================

INFO MAIN: Approved [node0 localhost]
INFO MAIN: Approved [node1 192.168.1.112]
INFO localhost: rendering (1, 90)
INFO 192.168.1.112: rendering (91, 180)
INFO localhost: Saved /tmp/0001.png
INFO localhost: Saved /tmp/0002.png
INFO 192.168.1.112: Saved /tmp/0091.png
INFO localhost: Saved /tmp/0003.png
.
.
INFO 192.168.1.112: Saved /tmp/0177.png
INFO localhost: Saved /tmp/0088.png
INFO 192.168.1.112: Saved /tmp/0178.png
INFO localhost: Saved /tmp/0089.png
INFO 192.168.1.112: Saved /tmp/0179.png
INFO localhost: Saved /tmp/0090.png
INFO localhost: I'm done, no more chunks left for me.
INFO 192.168.1.112: Saved /tmp/0180.png
INFO 192.168.1.112: I'm done, no more chunks left for me.

INFO MAIN: Done multiblendering.
INFO MAIN: Start time: 2014-03-14 19:11:06.489596
INFO MAIN: End time  : 2014-03-14 19:17:33.687683
INFO MAIN: Spent     : 0:06:27.198087
```

Salida render multiblender 2 VM

Al terminar la tarea de renderizado se obtiene un tiempo inferior al tiempo inicial. En este caso es de 6 minutos, 27 segundos y 199 milisegundos. En este escenario se observo el ancho de banda en la red local y se ven picos de las tareas que usaban la red.

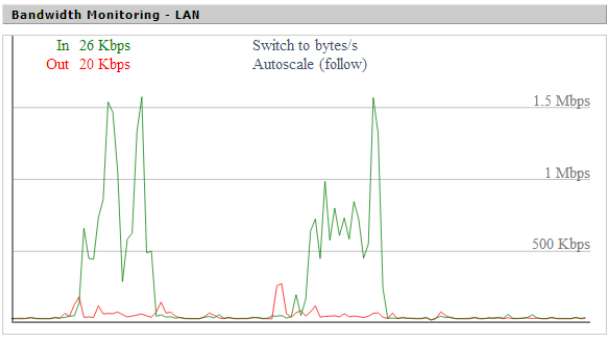


Figura 61 Consumo ancho de banda

c. *Tercera prueba 180 frames usando tres nodos*

Configuración archivo multiblendrc

```
root@GTMAIN:~# nano  ~/.multiblendrc
```

Edicion arhcivo multiblendrc

```
[main]
chunks=60
nodes=3
projectpath=/netrender/animacion
outputpath=/var/www/var/www/output
ssh=/usr/bin/ssh -Tq
scp=/usr/bin/scp -q
nice=0

[node0]
hostname=localhost
blender=/usr/bin/blender
workdir=/netrender/animacion
nice=0
ssh=/usr/bin/ssh -Tq
scp=/usr/bin/scp -q


[node1]
hostname=192.168.1.112
blender=/usr/bin/blender
workdir=/netrender/animacion
nice=0
ssh=/usr/bin/ssh -Tq
scp=/usr/bin/scp -q


[node2]
hostname=192.168.1.133
blender=/usr/bin/blender
workdir=/netrender/animacion
nice=0
ssh=/usr/bin/ssh -Tq
scp=/usr/bin/scp -q
```

Configuracion multiblendrc 3 VM

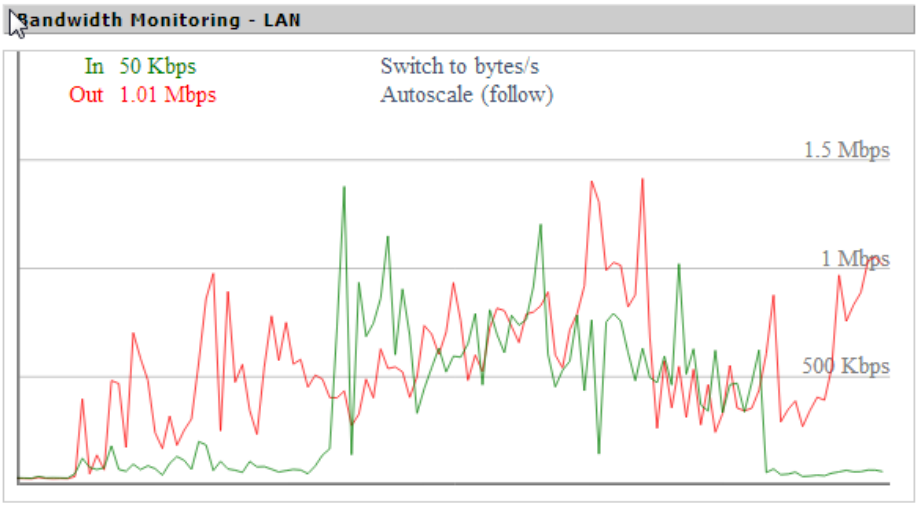
```
root@GTMAIN:~# multiblend -b /netrender/0015jos.blend -s 1 -e 180
```

Instrucción Multiblend

```
============================================================
Starting Multiblend 1.4
Created by Sybren A. Stüvel <sybren@stuvel.eu>
http://stuvel.eu/multiblend
============================================================

INFO MAIN: Approved [node0 localhost]
INFO MAIN: Approved [node1 192.168.1.112]
INFO MAIN: Approved [node2 192.168.1.133]
INFO localhost: rendering (1, 60)
INFO 192.168.1.112: rendering (61, 120)
INFO 192.168.1.133: rendering (121, 180)
INFO 192.168.1.112: Saved /tmp/0061.png
INFO localhost: Saved /tmp/0001.png
INFO 192.168.1.133: Saved /tmp/0121.png
INFO 192.168.1.112: Saved /tmp/0062.png
INFO localhost: Saved /tmp/0002.png
INFO 192.168.1.133: Saved /tmp/0122.png
INFO 192.168.1.112: Saved /tmp/0063.png
INFO localhost: Saved /tmp/0003.png
INFO 192.168.1.133: Saved /tmp/0123.png
INFO 192.168.1.112: Saved /tmp/0064.png
.
INFO localhost: Saved /tmp/0059.png
INFO 192.168.1.133: Saved /tmp/0179.png
INFO 192.168.1.112: Saved /tmp/0120.png
INFO 192.168.1.112: I'm done, no more chunks left for me.
INFO localhost: Saved /tmp/0060.png
INFO localhost: I'm done, no more chunks left for me.
INFO 192.168.1.133: Saved /tmp/0180.png
INFO 192.168.1.133: I'm done, no more chunks left for me.
INFO MAIN: Done multiblendering.
INFO MAIN: Start time: 2014-03-14 19:19:08.455391
INFO MAIN: End time  : 2014-03-14 19:23:36.739365
INFO MAIN: Spent     : 0:04:28.283974
```

Salida multiblend 3 VM

El tiempo obtenido fue de 4 minutos, 28 segundos y 284 milisegundos. Es posible observar que el uso de red se incremento en comparación con el segundo experimento.



Consumo de red 3 VM

d. *Cuarta prueba 180 frames usando cuatro nodos*

Configuración del archivo multiblenrc para cuatro VM.

```
root@GTMAIN:~# cat  ~/.multiblendrc
[main]
chunks=45 “se divide el total de frames por la cantidad de nodos”
nodes=4 “Se usaran 4 nodos para 180 frames”
projectpath=/netrender/animacion
outputpath=/var/www/output
ssh=/usr/bin/ssh -Tq
scp=/usr/bin/scp -q
nice=0

[node0]
hostname=localhost
blender=/usr/bin/blender
workdir=/netrender/animacion
nice=0
```

```
workdir=/netrender/animacion
nice=0
ssh=/usr/bin/ssh -Tq
scp=/usr/bin/scp -q


[node2]
hostname=192.168.1.133
blender=/usr/bin/blender
workdir=/netrender/animacion
nice=0
ssh=/usr/bin/ssh -Tq
scp=/usr/bin/scp -q

[node3]
hostname=192.168.1.138
blender=/usr/bin/blender
workdir=/netrender/animacion
nice=0
ssh=/usr/bin/ssh -Tq
scp=/usr/bin/scp -q
```

Configuracion para 4 VM

Se lanza la instrucción de renderizado

```
root@GTMAIN:~# multiblend -b /netrender/0015jos.blend -s 1 -e 180
```

Instrucción multiblender

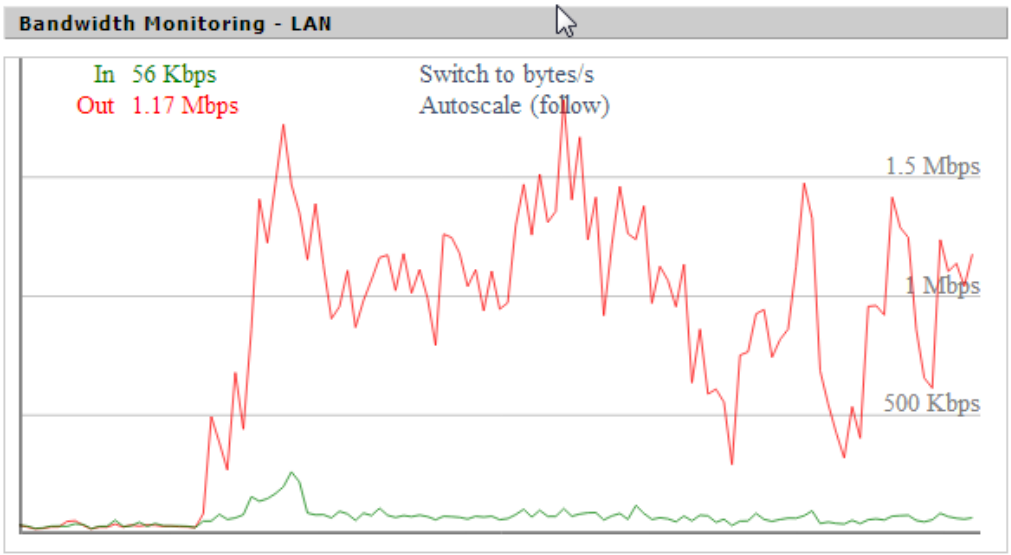
Proceso de renderizado en acción.

```
============================================================
Starting Multiblend 1.4
Created by Sybren A. Stüvel <sybren@stuvel.eu>
http://stuvel.eu/multiblend
============================================================

INFO MAIN: Approved [node0 localhost]
INFO MAIN: Approved [node1 192.168.1.112]
INFO MAIN: Approved [node2 192.168.1.133]
INFO MAIN: Approved [node3 192.168.1.138]
INFO localhost: rendering (1, 45)
INFO 192.168.1.112: rendering (46, 90)
INFO 192.168.1.133: rendering (91, 135)
INFO 192.168.1.138: rendering (136, 180)
```

```
INFO 192.168.1.112: Saved /tmp/0046.png
INFO 192.168.1.133: Saved /tmp/0091.png
INFO localhost: Saved /tmp/0001.png
.
INFO localhost: Saved /tmp/0043.png
INFO 192.168.1.133: Saved /tmp/0134.png
INFO 192.168.1.112: Saved /tmp/0089.png
INFO 192.168.1.138: Saved /tmp/0179.png
INFO localhost: Saved /tmp/0044.png
INFO 192.168.1.133: Saved /tmp/0135.png
INFO 192.168.1.133: I'm done, no more chunks left for me.
INFO 192.168.1.112: Saved /tmp/0090.png
INFO 192.168.1.112: I'm done, no more chunks left for me.
INFO 192.168.1.138: Saved /tmp/0180.png
INFO 192.168.1.138: I'm done, no more chunks left for me.
INFO localhost: Saved /tmp/0045.png
INFO localhost: I'm done, no more chunks left for me.
INFO MAIN: Done multiblendering.
INFO MAIN: Done multiblendering.
INFO MAIN: Start time: 2014-03-14 19:28:27.863988
INFO MAIN: End time  : 2014-03-14 19:31:59.247374
INFO MAIN: Spent     : 0:03:31.383386
```

Salida proceso renderizado 4 VM

El tiempo obtenido es de 3 minutos, 31 segundos y 384 milisegundos, el consumo de red es más alto que los experimentos anteriores



Consumo Ancho de banda 4 VM

9 Analizando los datos

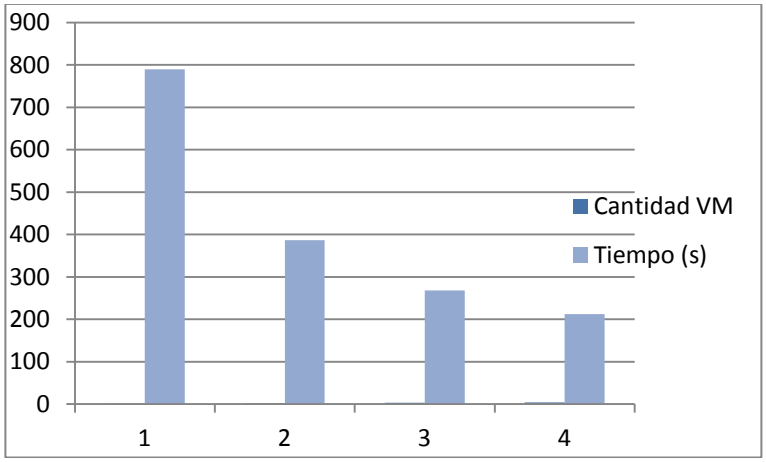
Para analizar y entender las ventajas de la computación en paralela es necesario analizar los datos obtenidos. Es importante tabular los datos usando diferente cantidad de VM

La siguiente tabal se debe construir en el laboratorio de computación distribuida, es importante aumentar la cantidad de nodos y graficar como se muestra a continuación.

Maquinas Virtuales						
Frames	Procesador	Memoria	Cantidad VM	Tiempo (s)	Recursos usados	Estándar
180	Intel Core i5	512	1	790,00	49%	1%
180	Intel Core i5	512	2	387,00	49%	1%
180	Intel Core i5	512	3	268,00	49%	1%
180	Intel Core i5	512	4	212,00	49%	1%

Tabulación de datos Experimento

El tiempo se unifico en segundos para poder visualizar la mejora de desempeño de la tarea distribuida. Las barras de la grafica representan el tiempo total de la tarea



Tiempo usado por cantidad VM

Una vez graficado es posible ver que la misma tarea disminuye su tiempo al aumentar la cantidad de VM. En la siguiente imagen es posible observar como hay

una tendencia que es inversamente proporcional a la cantidad de nodos. Es decir, a mayor cantidad de nodos menor tiempo de la tarea.

Tendencia de tiempo

Una vez se hayan analizado los datos se da final al Implementación De Un Laboratorio Para La Realización De Pruebas De Software Distribuido Y Procesamiento En Paralelo.

10 Como visualizar los resultados

Esta guía genera las de imágenes que se almacenan en el directorio "/var/www/output", sin embargo es posible ver que las imágenes se han creado al visitar la URL de la VM maestra. Es decir en un navegador de un equipo que esté conectado a la misma subred, es posible acceder vía web y visualizar los frames resultantes.

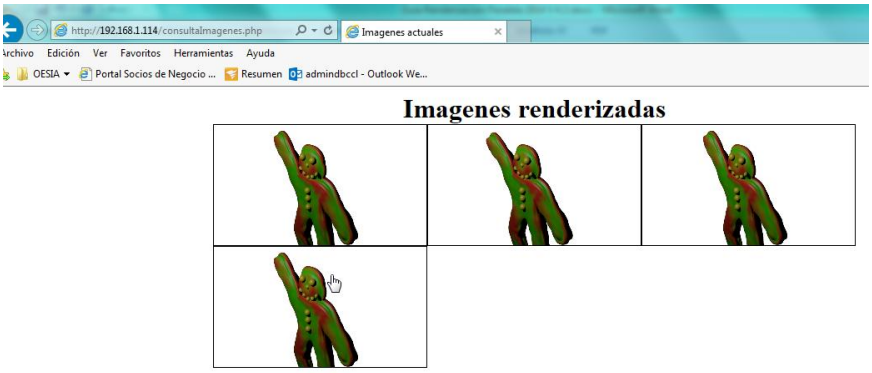
Ejemplo:

Si el comando "ifconfig" muestra la IP 192.168.1.114, es posible visitar esa IP en un navegador web agregando consultaImagenes.php al final de la misma. Es decir "192.168.1.114/consultaImagenes.php". recuerde cambiar la IP por la IP del laboratorio.

NOTA: La pagina que se muestra tiene el titulo de **Imágenes renderizadas.** Esta pagina tiene un control de ajax que refresca automáticamente y va mostrando las imágenes que se han renderizado.

Página de visualización de imágenes

Mientras se corre el experimento se irán mostrando las imágenes renderizadas, las cuales pueden ser descargadas para edición.



Visualización imágenes renderizadas

ANEXO B

El documento Encuesta Investigación Ver 2.2 consolidaba una encuesta que se pretendía realizar para comprobar un objetivo específico de esta guía.

Esta encuesta está orientada a validar las capacidades tecnológicas para un laboratorio de Grid en la sala de informática de la facultad de Ingeniería de acuerdo a los requerimientos de investigación.

1. ¿A cuál grupo de Investigación pertenece? Seleccione el grupo de investigación al que pertenece.
[A]. GISIC
[B]. GIP
[C]. GEGI
[D]. PAVIMENTOS Y MATERIALES GRANULADOS
[E]. AGUA Y SANEAMIENTO DE COMUNIDADES
[F]. NINGUNO
2. ¿Cuál es el problema de investigación sobre el cual trabaja? Describa su tema de investigación

3. ¿Qué software utiliza para el análisis de datos? Escriba el software, use comas si requiere escribir más de uno.

4 ¿Utiliza software para realizar simulaciones por computador?
Si
No

5 ¿Cual? Si respondió afirmativamente la pregunta anterior, por favor indique el/los nombre(s) del software de simulación que utiliza.

6 ¿Qué entiende por capacidad computacional?

7 ¿Le gustaría contar con un súper computador para realizar experimentos por simulación computacional?
Si
No

8 ¿Ha interactuado con otras áreas de la Universidad para el desarrollo de sus investigaciones?
Si
No

9. Si respondió afirmativamente a la pregunta anterior, por favor especifique con que unidades y en qué medida.

10 ¿Ha requerido computación en paralelo o distribuida para desarrollar sus investigaciones?
Si
No

Anexo C

Documento de referencia para tabular datos de los experimentos, al igual que preparar los comandos SSH-ADD y toma de datos para graficas del experimento.

VM	IP
LAB01	172.30.19.147
LAB02	172.30.18.144
LAB03	172.30.10.241
LAB04	172.30.19.241
LAB05	172.30.19.131
LAB06	172.30.28.58
LAB07	172.30.29.198
LAB08	172.30.11.180
LAB09	172.30.30.27
LAB10	172.30.18.32
LAB11	172.30.18.52
LAB12	172.30.18.45
LAB13	172.30.18.110
LAB14	172.30.17.83
LAB15	172.30.16.155
LAB16	172.30.16.168
LAB17	172.30.25.241
LAB18	172.30.22.5
LAB19	172.30.30.75
LAB20	172.30.30.29
LAB21	172.30.11.62
LAB22	172.30.25.75
LAB23	172.30.11.82
LAB24	172.30.10.145

Ejemplo de tabulación datos experimentos

ssh-copy-id root@	172.30.18.144
ssh-copy-id root@	172.30.10.241
ssh-copy-id root@	172.30.19.241
ssh-copy-id root@	172.30.19.131
ssh-copy-id root@	172.30.28.58
ssh-copy-id root@	172.30.29.198
ssh-copy-id root@	172.30.11.180
ssh-copy-id root@	172.30.30.27

ssh-copy-id root@	172.30.18.32
ssh-copy-id root@	172.30.18.52
ssh-copy-id root@	172.30.18.45
ssh-copy-id root@	172.30.18.110
ssh-copy-id root@	172.30.17.83
ssh-copy-id root@	172.30.16.155
ssh-copy-id root@	172.30.16.168
ssh-copy-id root@	172.30.25.241
ssh-copy-id root@	172.30.22.5
ssh-copy-id root@	172.30.30.75
ssh-copy-id root@	172.30.30.29
ssh-copy-id root@	172.30.11.62
ssh-copy-id root@	172.30.25.75
ssh-copy-id root@	172.30.11.82
ssh-copy-id root@	172.30.10.145

Ejemplo para preparación comando ssh-copy

VM Maestra	[main]
	chunks=180
	nodes=24
	projectpath=/netrender/animacion
	outputpath=/var/www/output
	ssh=/usr/bin/ssh -Tq
	scp=/usr/bin/scp -q
	nice=0
LAB01	[node0]
	hostname=172.30.18.144
	blender=/usr/bin/blender
	workdir=/netrender/animacion
	nice=0
	ssh=/usr/bin/ssh -Tq
	scp=/usr/bin/scp -q
LAB02	[node1]
	hostname=172.30.10.241
	blender=/usr/bin/blender
	workdir=/netrender/animacion
	nice=0
	ssh=/usr/bin/ssh -Tq
	scp=/usr/bin/scp -q

LAB03	[node2]
	hostname=172.30.19.241
	blender=/usr/bin/blender
	workdir=/netrender/animacion
	nice=0
	ssh=/usr/bin/ssh -Tq
	scp=/usr/bin/scp -q

Ejemplo configuración múltiples nodos