

UNIVERSIDAD DE HUÁNUCO
FACULTAD DE INGENIERÍA
PROGRAMA ACADÉMICO DE INGENIERÍA DE
SISTEMAS E INFORMÁTICA



TESIS:

“COMPUTACIÓN PARALELA: CLUSTER DE ALTO RENDIMIENTO VIRTUAL PARA EL INCREMENTO DE VELOCIDAD DE CALCULO DE PROGRAMAS”

PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO DE SISTEMAS E INFORMÁTICA

TESISTA

Tony Magiver, PIÑAN ROJAS

ASESOR

Ing. Edgardo Cristiam I., López De La Cruz

Huánuco- Perú

2018



UDH
UNIVERSIDAD DE HUÁNUCO
http://www.udh.edu.pe

UNIVERSIDAD DE HUÁNUCO

Facultad de Ingeniería

E.A.P. DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

ACTA DE SUSTENTACIÓN DE TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO(A) DE SISTEMAS E INFORMÁTICA

En la ciudad de Huánuco, siendo las 18:30 horas del día 24 del mes de JULIO del año 2018, en el Auditorio de la Facultad de Ingeniería, en cumplimiento de lo señalado en el Reglamento de Grados y Títulos de la Universidad de Huánuco, se reunieron el **Jurado Calificador** integrado por los docentes:

DR. RICHARD MARIN SEVILLANO (Presidente)
ING. PAOLO EDUER SOLIS JARA (Secretario)
ING. JOSÉ NUÑEZ VICENTE (Vocal)

Nombrados mediante la Resolución N°, para evaluar la **Tesis** intitulada:

" COMPUTACIÓN PARALELA: CLÚSTER DE ALTO RENDIMIENTO VIRTUAL PARA EL INCREMENTO DE VELOCIDAD DE CÁLCULO DE PROGRAMAS

presentado por el (la) Bachiller TONY MAGUER, PIÑAN ROJAS, para optar el Título Profesional de Ingeniero(a) de Sistemas e Informática.

Dicho acto de sustentación se desarrolló en dos etapas: exposición y absolución de preguntas: precediéndose luego a la evaluación por parte de los miembros del Jurado.

Habiendo absuelto las objeciones que le fueron formuladas por los miembros del Jurado y de conformidad con las respectivas disposiciones reglamentarias, procedieron a deliberar y calificar, declarándolo (a) APROBADO por UNANIMIDAD con el calificativo cuantitativo de 14 y cualitativo de SUFICIENTE (Art 47)

Siendo las 19:30 horas del día 24 del mes de JULIO del año 2018, los miembros del Jurado Calificador firman la presente Acta en señal de conformidad.

Presidente

Secretario

Vocal

Dedicatoria:

Este trabajo lo dedico a mi familia, de manera especial a mis padres Luz y Antonio quienes nunca dejaron de apoyarme y brindarme su amor sincero, sin ellos nada de esto seria posible, se lo dedico también a Aldair por darme la motivación para sacar adelante todos mis proyectos y luchar por alcanzar mis metas.

Agradecimientos:

Agradezco a mis padres por preocuparse de manera constante por mi desarrollo personal y profesional, gracias a mi universidad por brindarme las herramientas para convertirme en profesional en esta carrera que tanto me apasiona.

Finalmente agradezco a quien lea esta investigación, esperando que encuentre entre sus páginas la información relevante y puntual que busca.

ÍNDICE

DEDICATORIA.....	i
AGRADECIMIENTO.....	ii
INDICE.....	iii
INDICE DE TABLAS.....	v
RESUMEN	vi
ABSTRACT	vii
INTRODUCCIÓN.....	viii
CAPÍTULO I	9
PROBLEMA DE INVESTIGACIÓN	9
1.1. Descripción del problema.....	9
1.2. Formulación del problema	10
1.3. Objetivo General	10
1.4. Objetivos Específicos	10
1.5. Justificación de la investigación.....	11
1.6. Limitaciones de la investigación	11
1.7. Viabilidad de la investigación	12
CAPÍTULO II	14
MARCO TEÓRICO	14
2.1. Antecedentes de la Investigación	14
2.2. Bases Teóricas	22
2.3. Definiciones conceptuales.....	39
2.4. Hipótesis	39
2.5. Variables	40
2.5.1.Variable Independiente	40
2.5.2.Variable Dependiente.....	40
2.6. Operacionalización de Variables	40
CAPÍTULO III	41
METODOLOGÍA DE LA INVESTIGACIÓN	41
3.1. Tipo de Investigación	41
3.1.1.Enfoque.....	41
3.1.2.Alcance	41
3.1.3.Diseño.....	41
3.2. Población y Muestra.....	42
3.3. Técnicas e instrumentos de recolección de datos	42

3.4. Técnicas para el procesamiento y análisis de la información	42
CAPITULO IV	43
RESULTADOS	43
4.1 PROCESAMIENTO DE DATOS.....	43
4.2 CONTRASTACION DE HIPOTESIS Y PRUEBA DE HIPOTESIS	44
CAPÍTULO V	46
DISCUSIÓN DE RESULTADOS.....	46
CONCLUSIONES.....	48
RECOMENDACIONES.....	49
REFERENCIAS BIBLIOGRÁFICAS.....	50

INDICE DE TABLAS

Cuadro N.º 1: Estadísticas de muestras relacionadas	43
Cuadro N.º 2: Prueba de Kolmogorov-Smirnov para una muestra	44
Cuadro N.º 3: El ritual de la significancia estadística	45
Cuadro N.º 4: Resultados del número de repeticiones de las pruebas con uno y dos nodos	46

RESUMEN

La investigación se centra en la demostración a nivel aplicativo en un entorno virtual de la utilización de clústeres para la computación en paralelo, usando para ello un sistema operativo especialmente diseñado para el trabajo en malla, también se utiliza un programa que utilice las características del clúster y así ejecutar de forma más rápida los cálculos utilizando de dos a mas nodos conectados en una red de área local. Se llevó a cabo las pruebas en un entorno virtualizado con dos nodos con sus correspondientes sistemas operativos instalados, bajo el sistema operativo GNU/Linux, usando la distribución PelicanHPC. La muestra a tomar fue la ejecución de un programa que realiza un cálculo de forma repetida en seis veces reiteradas y continuas, primero usando un nodo y luego con dos nodos; el resultado fue satisfactorio ya que en las repeticiones usando los dos nodos el incremento de velocidad promedio disminuyo en 5 segundos, demostrando así que la computación paralela es efectiva al momento de usar la mismo tiempo la capacidad de procesamiento de cada nodo conectado en una red de área local. Los resultados de la ejecución del programa usando uno y dos nodos respectivamente varían entre 4 a 7 segundos de diferencia, incrementado la velocidad de procesamiento y disminuyendo el tiempo de ejecución del cálculo del programa.

Palabras clave: Computación paralela, clúster de alto rendimiento, PelicanHPC.

ABSTRACT

The research focuses on the demonstration at the application level in a virtual environment of the use of clusters for parallel computing, using an operating system specially designed for mesh work, also using a program that uses the characteristics of the cluster and thus execute more quickly the calculations using two or more nodes connected in a local area network. The tests were carried out in a virtualized environment with two nodes with their corresponding operating systems installed, under the GNU \ Linux operating system, using the PelicanHPC distribution. The sample to be taken was the execution of a program that makes a calculation of repeated form in six repeated and continuous times, first using a node and then with two nodes; the result was satisfactory since in the repetitions using the two nodes the average speed increase decreased by 5 seconds, demonstrating that parallel computing is effective when using the same time the processing capacity of each connected node in a network of local area. The results of the execution of the program using one and two nodes respectively vary between 4 to 7 seconds apart, increasing the processing speed and decreasing the execution time of the program calculation.

Keywords: Parallel computing, high performance cluster, PelicanHPC.

INTRODUCCIÓN

La presente investigación se desarrolló bajo el enfoque cuantitativo ya que se tomaron muestras previas y posteriores para las comparaciones correspondientes estos resultados fueron numéricos, expresados en segundos, relacionados al tiempo de cálculo de los programas usando un clúster de alto rendimiento. El objetivo principal es de instalar y configurar el sistema operativo GNU/Linux con la distribución PelicanHPC para la implementación del Clúster de alto rendimiento virtual para poder así demostrar el incremento de velocidad utilizando de dos a mas nodos en el clúster. Se trabaja con dos variables: Clúster de alto rendimiento y Velocidad de cálculo de los programas. La hipótesis formulada es: a utilización de un clúster de alto rendimiento virtual incrementará la velocidad de cálculo de programas. El nivel de la investigación se centró en el nivel aplicativo y de tipo tecnológico ya que se usa la tecnología para poder demostrar en forma práctica y en un entorno virtualizado el poder de procesamiento al usar dos o más nodos en un clúster de computadoras. El diseño a usarse es el pre experimental, en este caso no se cuenta con grupo de personas, si no se toman como muestras previas y posteriores los resultados obtenidos del tiempo de ejecución de los programas al realizar los cálculos usando el clúster, se realizaron cinco tomas cinco repeticiones del programa con sus correspondientes resultados expresados en segundos. En cuanto a la limitación principal fue el de no contar con los nodos físicos para poder hacer la aplicación en forma real y no virtual. Finalmente podemos concluir afirmando que con la utilización de un clúster de computadoras podemos incrementar la velocidad de procesamiento para el cálculo de programas y reducir el tiempo de ejecución del mismo, es así como en la presente investigación se reduce en 5 segundos el cálculo de un programa usando dos nodos en contraposición de usar un solo nodo.

CAPÍTULO I

PROBLEMA DE INVESTIGACIÓN

1.1. Descripción del problema

Existe una gran diferencia entre el avance y desarrollo del software y el hardware, podemos decir que la línea paralela entre estos dos, el software es la que tiene un mayor avance y desarrollo por lo tanto cada vez salen versiones más actualizadas del software necesitando así mayor capacidad en cuanto al poder de procesamiento y mayores requisitos en cuanto a hardware (memoria, discos, procesador, etc.).

En algunos casos las empresas se ven obligadas a adquirir nuevos equipos de cómputo para dar solución o respuesta a las demandas de los actuales softwares entre ellos software de aplicaciones, de base de datos, de cálculo y procesamiento entre otros, es así que las empresas con el tiempo descartan o desechan sus computadoras desfasadas reemplazándolas por nuevas, pero sin saber el poder de procesamiento que podrían tener al juntarlas y hacerlas trabajar al unísono en forma paralela. Es así que dentro de la ciencia de la computación existe una rama denominada Computación paralela la cual nos dice que podemos usar varios computadores al mismo tiempo todos conectados bajo una red y hacerlas trabajar en forma paralela aumentando el poder de procesamiento y cálculo.

A tal fin se desea dar a conocer o plantear la propuesta de reutilizar equipos de cómputo antiguos o desfasados utilizando la computación paralela en este caso un clúster de alto rendimiento para reutilizar dichos equipos y hacerlos trabajar conjuntamente todos conectados bajo una misma red.

En ese sentido se pretende dar a conocer y demostrar los beneficios de aplicar la computación paralela para reutilizar equipos de cómputos desfasados y mejorar la velocidad de procesamiento y cálculos en aquellos programas que demandan altos requerimientos en cuando a

velocidad de procesamiento. En la rama de la computación paralela emplearemos el clúster de alto rendimiento para tal fin, en este caso se usará el sistema operativo GNU/Linux para demostrar como trabajar un clúster de alto rendimiento y los efectos que produce al momento de realizar cálculos complejos en comparación de una máquina y de un conjunto de máquinas unidas bajo un clúster trabajando en forma paralela.

1.2. Formulación del problema

Formulación General

¿En qué medida se incrementa la velocidad de cálculo de programas empleando un clúster de alto rendimiento virtual?

Formulaciones Específicas

- A.** ¿Cuál es el procedimiento para Instalar y configurar el sistema operativo GNU/Linux con la distribución PelicanHPC para la implementación del Clúster de alto rendimiento virtual?
- B.** ¿De qué forma se realiza la comprobación del incremento de velocidad de cálculos los programas mediante el uso del clúster de alto rendimiento virtual?

1.3. Objetivo General

Demostrar que con la utilización del clúster de alto rendimiento virtual se incrementa la velocidad de cálculo de programas.

1.4. Objetivos Específicos

- A.** Instalar y configurar el sistema operativo GNU/Linux con la distribución PelicanHPC para la implementación del Clúster de alto rendimiento virtual

- B. Realizar la comprobación del incremento de velocidad de cálculo de los programas mediante el uso del clúster de alto rendimiento virtual.

1.5. Justificación de la investigación

1.5.1. Justificación Teórica:

La justificación teórica se basa en el hecho de contar con fuentes de información disponibles para el desarrollo, aplicación y pruebas de la parte técnica, correspondiente al presente estudio de investigación.

1.5.2. Justificación Práctica:

Mediante la presente investigación se da a conocer la técnica y/o método para implementar un clúster de alto rendimiento utilizando el sistema operativo GNU/Linux, así como también el procedimiento para la configuración y administración del mismo como también la ejecución de las pruebas.

1.5.3. Justificación Metodológica:

La metodología que se emplea se basa en los principios del software libre en el cual indica que podemos instalar las veces que queramos el sistema operativo, analizar su funcionamiento mediante su código, mejorarlo y así como también distribuir y difundir las mejoras.

1.6. Limitaciones de la investigación

Las limitaciones encontradas para realizar el presente estudio de investigación son:

- La parte aplicativa de la investigación será llevada a un escenario virtual, utilizando máquinas virtuales, las pruebas que se realizan en este escenario tal vez difieran con las pruebas ejecutadas en un entorno físico y real.

- ☑ No se realiza la aplicación y las pruebas en una organización o una empresa por lo tanto la propuesta y la demostración del incremento de velocidad de procesamiento usando el clúster de alto rendimiento es a nivel general usando escenarios virtuales, no se pretende dar solución a un problema específico para una determinada empresa u organización.

- ☑ Solo se utilizan dos programas de cálculos para las pruebas correspondientes, antes y después de la implementación del clúster de alto rendimiento. Se compara el rendimiento de la velocidad de ejecución de los programas con varias tomas y medidas para obtener un promedio de rendimiento.

- ☑ No se consideran como indicadores: tasa de transferencia de la red y la latencia porque las pruebas se hacen a nivel del cálculo del programa y se estima su velocidad de procesamiento en segundos, en base a un nodo y posteriormente a dos nodos.

1.7. Viabilidad de la investigación

1.7.1. Viabilidad Técnica.

El viable desde un punto de vista técnico, por contar con las herramientas necesarias para la implementación y pruebas del clúster de alto rendimiento, en este caso contar con la computadora conjuntamente con la máquina virtual, la red y el sistema operativo.

1.7.2. Viabilidad Económica.

El proyecto es viable económicamente porque se reutilizan los recursos que se cuenta en la investigación como computadoras y dispositivos, así mismo el uso del software es gratuito ya que

se opta por emplea software libre tanto para la implementación y las pruebas del estudio.

1.7.3. Viabilidad Recursos Humanos.

El proyecto es viable desde la perspectiva de recursos humanos ya que se cuenta con el investigador encargado de la realización del proyecto ejecución y redacción del informe final bajo la supervisión de un asesor especializado en el tema.

1.7.4. Viabilidad Ética y Moral.

Las pruebas y ejecución del proyecto se realizan bajo un entorno virtualizado usando software libre, por lo tanto, no se utiliza software pirata y no licenciado para la ejecución del proyecto.

CAPÍTULO II MARCO TEÓRICO

2.1. Antecedentes de la Investigación

A. A nivel Internacional:

(CÁCERES, 2012), *Estrategia de implementación de un clúster de alta disponibilidad de n nodos sobre Linux usando software libre.*

Para optar el grado de Técnico en Ingeniería de Sistemas.

Los puntos más resaltantes fueron:

Con la realización de este trabajo se definió una estrategia para la implementación de un clúster de alta disponibilidad de N nodos usando software libre, que puede ser usada para proveer redundancia a las aplicaciones críticas de una empresa. A través de este trabajo, también fue posible proveer un sistema de tolerancia a fallas a servidores Linux usando un clúster de Alta Disponibilidad, cumpliendo de esta manera una de las metas planteadas al inicio del proyecto.

En el desarrollo teórico del proyecto se definieron los componentes básicos para la implementación de un clúster HA y se explicaron los aspectos más importantes de cada uno, además se mencionó la función que tiene cada componente y como se integran dentro del clúster de Alta Disponibilidad.

Con la finalidad de demostrar que la estrategia planteada si funciona, primero se diseñó e implementó un clúster de alta disponibilidad con 2 nodos para ofrecer los servicios de FTP y DNS. Luego se añadió un nodo adicional para ofrecer el servicio de HTTP y se modificó el diseño original de tal manera que los recursos sean distribuidos a nodos específicos dentro del clúster y que los servicios de FTP y DNS estén siempre juntos en un mismo nodo; logrando así un ejemplo de implementación para un escenario avanzado.

Durante la implementación del prototipo del proyecto también se indicó paso a paso cómo instalar, configurar, integrar y probar el software necesario para formar un clúster de alta disponibilidad. Así mismo se mostró cómo modificar la configuración del clúster para agregar un nodo adicional y se dio una breve explicación de los comandos más útiles para la administración del clúster.

Es importante mencionar que pese a que los problemas encontrados durante la instalación de los componentes para formar el clúster fueron solucionados probando diferentes versiones del software, los problemas encontrados durante la configuración e integración de cada componente demandaron más lectura e investigación que la esperada, sobre todo por problemas de incompatibilidad y errores de configuración ya que la información encontrada en los documentos oficiales, libros y artículos en Internet está desactualizada e incompleta, incluso fue necesario leer el código fuente de algunas aplicaciones.

(GONZÁLEZ RAMÍREZ EDGAR RUBÉN y RODRÍGUEZ SÁNCHEZ ABIMAEL, 2008), *Estrategia diseño e implementación de un clúster tipo beowulf para el desarrollo de cómputo científico avanzado*.

Para optar el grado de Técnico en Ingeniería en Comunicaciones y Electrónica.

Los puntos más resaltantes fueron:

Se realizaron las instalaciones de paquetes necesarios, así como las pruebas para el diseño de una plataforma en base a la consideración de algunos diseños que ya existen en ciertas arquitecturas, pero los cuales debieron ser optimizados a la arquitectura de procesador con el que contábamos. Esto es muy diferente a lo que ya se tiene hecho debido a que los procesadores de 64 bits apenas están siendo introducidos en el mercado. Es por ello que la determinación del sistema operativo más viable a usar,

instalación de paquetes, tipo de red a utilizar y aplicación a 64 bits se necesitó evaluar para este proyecto.

Podemos decir básicamente que las arquitecturas de computadoras que hoy en día existen dejan mucho que desear, optando por realizar algunas modificaciones para poder obtener un mejor desempeño. La primera conclusión importante en el uso de procesadores de 64 bits, es que no quiere decir que sea mucho más rápido un procesador de 64 bits o el doble de rápido que tener un procesador de 32 bits. La diferencia principal que se puede observar, es que sí se tiene un aumento de velocidad. Pero, la principal ventaja se obtiene en el hecho de que por tener un procesamiento de palabra de 64 bits cuando los procesos son muy grandes en número y en capacidad, se puede apreciar que los procesadores de 64 bits pueden seguir trabajando con muchas aplicaciones o con una aplicación muy demandante; mientras que los procesamientos de palabra de 32 bits simplemente no pueden realizar la aplicación. Aunque un mismo proceso pequeño podría ser ejecutado en ambos procesadores en el mismo tiempo.

Muchas veces como usuarios no sabemos que es lo que compramos. Y nos referimos a que a través del uso de multicore pudimos observar que no nos sirve de mucho tener 1 procesador con 4 núcleos, si las aplicaciones que usualmente utilizamos son ejecutadas de forma serial. Son muy pocas las aplicaciones que utilizan el paralelismo. Ejemplo de estas podrían ser algunos videojuegos y cualquier tipo de aplicación virtual por lo que sería innecesario tener varios núcleos. Lo único que estaríamos haciendo en el caso de que no se tengan aplicaciones paralelas es pagar por una tecnología que no estamos usando. Se muestra en este proyecto el objetivo de una solución que resuelve el paradigma del cómputo paralelo por medio de hardware con características especiales, que al agruparse cada uno de los núcleos, se puede maximizar su rendimiento considerablemente, ofreciendo capacidades de procesamiento de alto rendimiento

que no podría brindar de manera individual debido a sus limitaciones desde su diseño, ya que son equipos de cómputo que no están diseñados propiamente al procesamiento de alto rendimiento. Otra característica que se resalta en esta solución es su modularidad, es decir se pueden agregar nodos o equipos de diferentes arquitecturas sin que esto decremente su rendimiento, sino por el contrario la hace mucho más poderosa en su rendimiento y capacidades de procesamiento; y no lo limita o la hace exclusivo en cuanto a la configuración del hardware a agregar; de este modo, es un arreglo flexible y económico que aprovecha tecnologías tanto actuales como de generaciones anteriores en su composición física. Al no contarse con una aplicación específica para el clúster las pruebas a realizarse se limitaron únicamente a verificar su funcionamiento y comunicación entre los nodos que lo conforman.

Estas pruebas arrojaron resultados positivos por lo cual la hipótesis de crear una plataforma dedicada funcional para el cómputo paralelo a partir de equipo con capacidad de procesamiento de alto desempeño (Quad Core), utilizando software libre es correcta debido a que se obtuvieron mejoras en el tiempo de ejecución de algunos programas paralelos. Mejoras que son muy considerables si trata de cálculos muy grandes, en los que se requiera quizás cálculos de matrices muy grandes. No cabe duda que si Esto permite entregar un equipo que sirva como herramienta a la línea de investigación “Cómputo paralelo aplicado al análisis electromagnético utilizando el método de diferencias finitas en el dominio del tiempo”

(Martínez, 2004), *Implementación de un servidor web virtual Balanceador de carga basado en Linux.*

Para optar el grado de Magister en ciencias Telemáticas.

Los puntos más resaltantes fueron:

No es fácil construir un clúster debido a que los manuales no tienen todos los puntos a seguir para la construcción los cuáles

omiten muchos pasos. Antes de construir algo se tiene que leer lo suficiente para poder empezar con una idea de cómo se construye.

El clúster necesita un hardware idóneo, como pueden ser las tarjetas de red y el concentrador de mayor velocidad u otro tipo de conexión, para mejorar velocidad de respuesta y transferencia del clúster, otro tipo de dispositivo son los discos duros, la memoria RAM, entre otras cosas. La unidad de respaldo de energía, No break, es importante debido a que una falla de energía externa puede desconfigurar el clúster, cuando se apaguen las computadoras y funcione al 50% debido al fallo de corriente. La manera de distribuir las peticiones es eficiente entre las otras máquinas del clúster. Existe un programa para visualizar el estado del clúster como mencionamos en los capítulos anteriores llamado Piranha. Con este programa nos damos cuenta si el clúster está funcionando y como está distribuyendo la carga de las peticiones y ver las conexiones activas.1 inciso A se muestra el número de máquinas o servidores reales que se encuentran funcionando en el clúster, y sus direcciones IP internas, en el inciso B muestra que los servidores reales no tienen ninguna petición lo cual significa que el clúster está inactivo, en el inciso C vemos los procesos LVS que son los componentes de clúster ipvsadm, nanny y pulse, el clúster tiene un tiempo de espera para responder a las peticiones y si no encuentra ninguna se pone es un estado de inactivo.

B. A nivel Nacional:

(BULEJE, 2014), *Influencia de un clúster de computadoras de alto rendimiento en el tiempo de renderización de modelos 3d foto realistas, universidad nacional José María Arguedas 2013.*

Para optar el título Profesional de Ingeniería de Sistemas.

Los puntos más resaltantes fueron:

Al finalizar la exploración tecnológica de la investigación se llegó a la conclusión de que existe una amplia y muy variada colección de programas y paquetes de software libre disponibles que pueden utilizarse para implantar soluciones de clúster de computadoras de alto rendimiento, se concluyó la exploración tecnológica con la comparativa sobre estas diversas soluciones y la elección del sistema Clusterknoppix. Por otro lado también se realizó la exploración tecnológica sobre las herramientas para renderizar escenas tridimensionales con trazado de rayos, de la misma manera se realizó una comparativa entre las diversas soluciones encontradas y se escogió las herramientas Povray y Povmosix para realizar el trazado de rayos.

Al concluir el estudio realizado sobre los factores a tomar en cuenta para determinar los tiempos de procesamiento en la renderización de modelos tridimensionales foto realistas utilizando trazado de rayos se concluyó que los factores se clasifican en dos grandes grupos, por un lado se halla las estadísticas con respecto a información del modelo, el cual contiene la cantidad de objetos, luces, rayos entre otros, por otro lado se hallan los tiempos de renderización propios, se tomó solo el tiempo de renderización total por ser el más importante y suficiente para determinar la incidencia de mejora.

Se logró implantar un clúster de computadoras de alto rendimiento en el laboratorio 4 de la escuela profesional de ingeniería de sistemas de la Universidad Nacional José María Arguedas, el clúster utiliza el sistema Clusterknoppix junto con su monitor de recursos Openmosixview el cual permite evaluar y obtener los recursos computacionales del clúster.

Se evaluó los tiempos de renderización con la renderización de 384 escenas tridimensionales tomadas al azar del modelo benchmark con la utilización del clúster de computadoras de alto rendimiento y el software Povmosix, asimismo se determinó los

porcentajes de incidencia de mejora éntrelos diversos grupos experimentales de la investigación.

Los clúster de computadoras de alto rendimiento reducen drásticamente los tiempos de renderización de modelos y escenas tridimensionales foto realistas utilizando la técnica de trazado de rayos, la influencia de mejora se puede visualizar claramente cuando se utiliza un gran número de ordenadores, más por el contrario si se utiliza unos pocos ordenadores la incidencia de mejora es muy poca, pero a medida que se incrementa una buena cantidad de nodos al clúster se puede ver muy notablemente esta diferencia en los tiempos de renderización llegando a ser considerada como drástica con un 82.77%, pero llega también el punto en el que no se puede reducir más el tiempo de renderización al ir incrementado muchos más nodos esto se debe a que la coordinación de los trabajos se vuelve más complicada , por ende los tiempos de renderización disminuyen en muy poca proporción llegando la reducción mínima de nuestro caso de estudio en 2.00% entre el grupo experimental 1 y 2 una cifra no tan notable.

A lo largo de todo este tiempo el hombre ha ido construyendo súper ordenadores que le ayuden a resolver problemas cuya complejidad computacional suele ser elevada, los clústeres de computadoras de alto rendimiento son prueba de ello, la solución en clúster aún se mantiene vigente y es utilizada en el mundo de la computación gráfica, así como en otros sectores para acelerar los cálculos como se muestra en el anexo 5(Top 500 supercomputers).

(Zamora, 2014), *Clúster de servidores Linux para alta disponibilidad de la información.*

Para optar el título Profesional de Ingeniería de Sistemas.

Los puntos más resaltantes fueron:

La necesidad de alta disponibilidad en el caso de estudio es evidente al dar un servicio de información a un sistema de vital

importancia en las operaciones de la empresa, por lo cual fue necesario analizar y ver en mercado las mejores opciones y alternativas para el diseño e implementación del clúster que brinde

la alta disponibilidad requerida. El diseño e implementación del clúster se dio en un entorno Linux, el cual es muy complejo en comparación con un entorno Windows, para lo cual es necesario tener conocimiento de lenguajes de programación y códigos establecidos para la consola que fue la principal herramienta para la implementación del clúster ya que por medio de esta se accede a los archivos y variables de configuración en la implementación.

Los resultados obtenidos en el análisis de datos nos dan a relucir que la implementación del clúster brinda una disponibilidad de la información además de un mejor performance en el rendimiento del servicio teniendo características fundamentales para el óptimo desempeño del clúster como la escalabilidad y el balanceo de carga. La escalabilidad y el balanceo de carga permiten a nivel de hardware tener un mayor tiempo de vida de los equipos ya que la carga del tráfico de datos y las transacciones enviadas al servidor se distribuyen entre los nodos que lo componen. La arquitectura del clúster ha permitido alcanzar un aproximado de 99.99% de disponibilidad al año incluido las interrupciones previstas por mantenimiento de cada uno de los nodos que conforman el clúster.

C. A nivel local:

- ❖ Se realizaron las búsquedas en los repositorios de las Universidades e institutos de la ciudad y no encontraron trabajos de investigaciones similares al trabajo de investigación presente.

2.2. Bases Teóricas

Computación Paralela

Para (Dormido, S., Hernández, R., Ros, S., & Sánchez, J., 2003), la computación paralela o procesamiento en paralelo consiste en acelerar la ejecución de un programa mediante su descomposición en fragmentos que pueden ejecutarse de forma simultánea, cada uno en su propia unidad de proceso. Surge así de forma natural, la idea de la computación paralela, que genéricamente consiste en utilizar n computadoras para multiplicar, idealmente por n la velocidad computacional obtenida de un único computador. Por supuesto esta es una situación ideal que muy rara vez se consigue en la práctica.

Normalmente los problemas no pueden dividirse perfectamente en partes totalmente independientes y se necesita, por tanto una interacción entre ellas que ocasiona una disminución de la velocidad computacional. En este sentido se habla de mayor o menor grado de paralelismo en la medida de que un algoritmo sea más o menos divisible en partes independientes con igual costo computacional.

La topología que se empleó para la prueba del clúster fue estrella que los nodos de red (pc) estaban conectados a un switch central.

En cuanto a la arquitectura de red se usa Fast Ethernet donde los datos se transmiten a una velocidad teórica máxima de 100 Mbps.

Clúster de computadoras

Historia

La historia de los clúster de computadoras comenzó cuando Donald Becker y Thomas Sterling (1994) construyeron un clúster para la NASA, su nombre fue Beowulf con la finalidad de resolver problemas que aparecen en las ciencias de la tierra y del espacio.

Posteriormente en el año 1996, se construye otro clúster llamado Hyglac desarrollado por el Instituto Tecnológico de California (Cal Tech) y el Laboratorio de Propulsión Jet (JPL), el otro clúster es, Loki construido en el Laboratorio Nacional de los Álamos.

A partir de este proyecto, han surgido numerosas iniciativas en este sentido. Estos clúster se utilizan para cualquier tarea que requiera enormes cantidades de cómputo.

En la actualidad, los clúster han tenido mucho auge en centros de investigación y en las empresas, debido a que ciertos problemas que se deseaban resolver rebasaban la capacidad de cómputo de una computadora personal. La solución más obvia para resolver estos problemas pues sería comprar una supercomputadora, pero existe un problema a esta solución, debido a que una supercomputadora en ocasiones cuesta varios miles de dólares, cantidad que a veces va más allá de los presupuestos de inversión tanto de las empresas como de los centros de investigación.

Pero la necesidad de solucionar los problemas con los recursos que se cuentan, han hecho que personal académico de diversas universidades y centros de investigación se han a la tarea de construir sus propias supercomputadoras conectado computadoras personales y desarrollando software para resolver dichos problemas.

Definición de Clúster

Existen varias definiciones de clúster las cuales se asemejan mucho, se tomará como referencia algunas definiciones de distintos autores.

Según (Branch Bedoya, J. W., & Mesa Munera, A., 2008) Un clúster se puede definir como un grupo de múltiples computadores unidos por una red de alta velocidad de tal forma que el conjunto puede ser visto como una única máquina, pero que por su poder de cómputo resuelve problemas que un solo equipo de escritorio no podría hacer.

Según (Sánchez, E., & Heider, Y., 2007), un clúster es un conjunto de computadoras interconectadas con dispositivos de alta velocidad que actúan en conjunto usando el poder cómputo de varios CPUs en combinación para resolver ciertos problemas dados.

Se usa un clúster para crear una supercomputadora que puede servir como un servidor en un sistema, reduciéndose el costo de inversión

Según (Sterling, 1999), un clúster es la interconexión de dos o más computadoras independientes a través de una red, usadas como un recurso unificado de cómputo con el fin de aumentar el rendimiento en la ejecución de tareas.

Analizando las definiciones mencionadas anteriormente un clúster surge a necesidad de resolver problemas complejos pero que sí los puede realizar un equipo de cómputo en un lapso de tiempo muy largo, claro dependiendo de la complejidad del trabajo encomendado.

Un clúster de computadoras puede utilizar diversas tecnologías de interconexión y software de comunicación, seleccionar una tecnología de interconexión de red depende de diversos factores, la compatibilidad con el hardware del clúster, así como el sistema operativo y por ende el rendimiento, hay dos métricas para medir el rendimiento para interconexiones. El ancho de banda y la latencia, el ancho de banda está definido como la cantidad de datos que se puede transmitir a través del hardware de red en un determinado tiempo el cual se mide en Mbyte/s, mientras que latencia es el tiempo para preparar y transmitir datos de un nodo origen a un nodo destino que se mide en μ s.

Para los clúster de cómputo de pequeña escala, la mayoría de los constructores de clúster por primera vez tienden a elegir entre un bajo costo e interconexión de los productos básicos (Fast Ethernet o Giga bit Ethernet) o una interconexión más cara pero más alta en

rendimiento (como Myrinet o SCI). En muchos casos, esta elección se hace antes de cualquier investigación de la interconexión, necesidades de las aplicaciones que se ejecutan en el clúster y es a menudo determinado simplemente haciendo la pregunta: "¿Puede nuestro presupuesto permitirse una interconexión rápida?" Este enfoque no es el deseable y puede conducir a la frustración con el rendimiento del clúster resultante debido a dos razones posibles.

- Una posible razón es que la interconexión de la red seleccionada no sea adecuada para el trabajo a realizar.
- Otra posible razón es que el dinero gastado en una interconexión podría haber sido mejor gastado en otros componentes útiles, tales como procesadores más rápidos y de memoria más grande.

Los constructores de clúster también podrían fácilmente evitar estos problemas siguiendo algunas pautas simples y teniendo en cuenta el rendimiento y características de las diversas interconexiones según lo determinado por análisis independientes.

En general, el costo de los productos básicos de interconexión es menor que el costo de interconexiones de alto rendimiento. La principal diferencia entre los productos básicos e interconexiones de alto rendimiento es la latencia de envío de mensajes y el ancho de banda disponible para la mensajería. Esto significa, naturalmente, que interconexiones de alto rendimiento son más deseables cuando las aplicaciones que se ejecutan sobre ellos se comunican con frecuencia (sobre todo si se intercambian muchos mensajes pequeños). Las aplicaciones que se comunican con poca frecuencia y que intercambian grandes mensajes suelen realizar muy bien el uso de los productos básicos de interconexiones. El costo de Fast Ethernet (100Mbps) es ahora tan baja que es una estrategia común entre muchos constructores de clúster comenzando por utilizar Fast Ethernet para su interconexión y luego actualizar cuando sea necesario (o cuando tengan una mejor idea de sus características de aplicación y por lo tanto los requisitos). Dado el bajo coste de Fast

Ethernet, hay poca preocupación si el equipo se utiliza sólo durante un corto período de tiempo.

Características de un clúster

“Un clúster es un sistema distribuido por lo tanto posee sus características, pero adicionalmente proporciona tres beneficios principales: alta disponibilidad, buena escalabilidad y bajo costo”. (Sun, 2006).

- Alta disponibilidad

La disponibilidad es la capacidad de que un clúster pueda permanecer funcionando ante diferentes fallas que pudieran existir mientras se encuentra trabajando. Por ejemplo, una falla podría ser un procesador o equipo de cómputo estropeado, una falla de disco duro, alguna falla en la conexión de red o falla de energía eléctrica en algún nodo, entonces si un nodo falla los demás nodos captan toda la carga de trabajo.

Para muchas aplicaciones de empresas, el clúster es una buena solución, debido a que el tiempo de recuperación es aceptable. Frecuentemente se programan los tiempos para realizar otras aplicaciones en algún nodo del clúster.

- Buena escalabilidad

La escalabilidad es la capacidad de un sistema o sistemas de poder crecer de acuerdo con las necesidades de los usuarios. Los clúster son escalables ya que se les puede agregar o sustraer componentes a la computadora, por ejemplo, agregar memoria, discos duros, tarjetas de red, procesadores o agregar otra computadora.

Las arquitecturas de los clúster pueden ser escalables de dos maneras. La primera, los diseñadores seleccionan las computadoras que formarán los nodos del clúster, para que de

esta forma puedan ser escalables, aumentándoles memoria, disco duro, etc. La segunda, los diseñadores aumentan el número de computadoras que forman parte del clúster.

- **Bajo costo**

Una arquitectura de clúster es relativamente bajo en costo debido a que un clúster utiliza en el mayor de los casos software libre y gratuito, de esta manera las empresas no tendrían que hacer gasto extra para comprar software, también es importante señalar y especificar que se puede utilizar los mismos equipos de cómputo que posea la organización para construir el clúster sin la necesidad de comprar equipos adicionales.

Ventajas y desventajas de los clúster

Las ventajas que presentan la construcción de un clúster citadas por (Torrealba Martinez, 2002) son:

- Cada una de las máquinas en un clúster, puede ser un sistema completo para usarlo en un amplio rango de aplicaciones.
- El hardware de interconexión de red ha venido experimentando un constante decremento de precio, considerando que además se puedan lograr ahorros adicionales empleando un solo monitor, teclado y ratón.
- Los clúster de computadoras pueden crecer hasta formar sistemas verdaderamente grandes, es decir, desde dos hasta varios cientos de nodos.
- Se puede reemplazar fácilmente una computadora del clúster que no esté funcionando correctamente.
- Si algún componente falla, el o los procesos pueden seguir ejecutándose en los demás nodos.

- El clúster se puede interconectar a una red de área local permitiendo dar servicio a múltiples usuarios internos y externos a través de internet.
- Existe software gratuito en la red como el sistema operativo para computadoras de escritorio como GNU/LINUX y software de aplicación para clúster como: MOSIX, CONDOR, LUI, BEOULF, HOVM, entre otros.

Las desventajas principales en la utilización de un clúster según son:

- Si falla el nodo maestro desde el cual se envía una tarea, se pierde la ejecución de la tarea.
- Puede haber inconsistencia en los datos, provocada por la falla en el software o en el hardware.
- Una pregunta puede surgir entonces, si el software es gratis, el hardware barato y los clúster pueden crecer de manera significativa. ¿Por qué cualquier empresa o institución no tiene su propio clúster?, pues bien no todo hardware de red está diseñado para realizar procesamiento paralelo, generalmente por que la latencia es alta y el ancho de banda de la red es baja, aún hay muy poco software que soporte a un clúster como un solo sistema.

Lenguajes de programación para clúster

“Existen varios lenguajes de programación paralela, sobresaliendo de estos MPI (Message Pasing Interface) – Interface de paso de mensajes y PVM (Paralelo Virtual Machine)- máquina virtual paralela, por ser los estándares más aceptados.” (Castillo Castillo, J., & Olivera Acosta, R. B., 2006).

- MPI

MPI consiste de una biblioteca para programación paralela en el modelo de intercambio de mensajes. En este estándar se han incluido los aspectos más relevantes de otras bibliotecas de programación paralela.

Entre las ventajas de MPI se encuentra la disponibilidad de varios modos de comunicación, los cuales permiten al programador el uso de buffers para el envío rápido de mensajes cortos, la sincronización de procesos o el traslape de procesos de cómputo con procesos de comunicación. Esto último reduce el tiempo de ejecución de un programa paralelo, pero tiene la desventaja de que el programador debe ser más cuidadoso para evitar la corrupción de mensajes. Dos de las principales distribuciones libres de MPI son: LAM/MPI y MPICH.

- PVM

PVM se comenzó a desarrollar en verano de 1989 por el Oak Ridge National Laboratory, y posteriormente junto con la Universidad Tennessee en los EUA. Es una biblioteca de envío de mensajes, totalmente libre, capaz de trabajar en redes homogéneas y heterogéneas y que hacen uso de los recursos existentes en algún centro de trabajo para poder construir una máquina paralela de bajo costo, obteniendo su mejor rendimiento.

Maneja transparentemente el envío y recepción de todos los mensajes, conversión de mensajes y calendarización de tareas a través de una red de arquitecturas incompatibles. Está diseñado para conjuntar recursos de cómputo y proveer a los usuarios de una plataforma paralela para correr sus aplicaciones, independientemente del número de computadoras distintas que utilicen y donde estas se encuentren localizadas. El modelo computacional de PVM es simple y además muy general. El usuario escribe su aplicación como una colección de tareas cooperativas. Las tareas acceden los recursos de PVM a través de una biblioteca de rutinas. Estas rutinas permiten la inicialización y terminación de tareas a través de la red, así como la comunicación y sincronización entre tareas.

Modelos de clúster

En los artículos publicados por (Branch Bedoya, J. W., & Mesa Munera, A., 2008) existen tres modelos de clúster Mosix / Openmosix y Beowulf.

- Beowulf

“Beowulf es una clase de computador masivamente paralelo de altas prestaciones principalmente construido a base de clúster de componentes hardware estándar”. (Llorens, E., & Peña, M., s.f.)

“Consta de un conjunto de nodos minimalistas, unidos por un medio de comunicaciones económico. Esto quiere decir que tienen lo mínimo para ejecutar su función, de hecho, los nodos por si solos no son capaces de ejecutar ni tan siquiera un sistema operativo” (Arrollo, R., Nievas, F., & Pino, O. (, 2004)

Se puede ver como un supercomputador paralelo construido con hardware comercial de fácil adquisición, que posee como sistema operativo (GNU/Linux)

“Los clústers Beowulf son extremadamente poderosos, pero no lo son para todas las personas. Su principal desventaja es que requieren de software diseñado para poder aprovechar los recursos del clúster” (Robbins)

- Mosix

Según (Pérez, 2001) la arquitectura Mosix basada en la misma ideología de la arquitectura Beowulf. Se basa en un conjunto de parches aplicados al kernel de Linux y que se asignan a todo un grupo de nodos un espacio de direcciones y de procesos común, gracias a los cuales los procesos migran de uno a otro con el fin de equilibrar favorablemente la carga del sistema global.

“Es una herramienta diseñada para realizar balanceo de carga en el clúster de forma totalmente transparente de manera tal que los

nodos se comportan como un asola máquina, y así incrementar el aprovechamiento de cada uno de los nodos” (Correa, 1999)

La principal ventaja de Mosix frente a Beowulf se basa precisamente en esto. Mosix da mejor respuesta que Beowulf frente a la caída e inserción de nodos. A parte de esto, Mosix permite un cambio constante de aplicación, o incluso, el correr aplicaciones independientes de forma simultánea.

- Openmosix

Openmosix es una extensión del proyecto Mosix, De (Chirinov, 2004), la idea de este modelo es que la distribución de tareas en el clúster la determina Openmosix de forma dinámica, conforme se van creando tareas. Cuando un nodo está demasiado cargado, las tareas que se están ejecutando pueden migrar a cualquier otro nodo del clúster. Así desde que se ejecuta una tarea hasta que se muere, podrá migrar de un nodo a otro, sin que el proceso sufra mayores cambios.

La idea de Openmosix es que los procesos colaboren de forma que parezcan que están en un mismo nodo. Como sus algoritmos son dinámicos, tiene una fuerte ventaja sobre los algoritmos estáticos de PVM/MPI, responden a las variaciones en el uso de los recursos entre los nodos migrando procesos de un nodo a otro, de forma transparente para el proceso, para balancear la carga y para evitar fallas de memoria en un nodo.

Al contrario que PVM/MPI, no necesita una adaptación de la aplicación, ni siquiera que el usuario sepa nada sobre el clúster. Para aprovechar completamente PVM/MPI hay que programar con sus librerías, por lo tanto hay que rehacer todo el código que haya para sacar el mayor provecho del clúster.

Openmosix puede balancear una única aplicación si esta está dividida en procesos lo que ocurre en un gran número de aplicaciones hoy en día, también puede balancear las aplicaciones

entre sí, balanceando los procesos en la mínima unidad de balanceo.

Además, Openmosix funciona a nivel kernel por lo tanto puede conseguir toda la información que necesite para decidir cómo está de cargado un sistema y que pasos debe seguir para aumentar el rendimiento, además puede realizar más funciones que cualquier aplicación a nivel de usuario.

Aplicaciones de clúster

Entre las aplicaciones más recientes de la computación distribuida se encuentra la computación en la nube o cloud computing que todos usamos en el acceso a múltiples servicios de internet.

La aplicación de estas técnicas se aplica a dominios muy distintos que tienen en común a necesidad de una gran potencia de cómputo. Algunas aplicaciones tradicionales incluyen la predicción del tiempo, las simulaciones aerodinámicas el estudio de los terremotos y tsunamis o la secuenciación del genoma humano.

Por ejemplo, en los recientes estudios sobre el Bolson de Higgs han tenido un papel central el uso de grandes computadores las técnicas de procesamiento paralelo.

La computación paralela y distribuida también se usa en el estudio en las interacciones entre proteínas y el modelado de moléculas que tienen una gran relación con el diseño de nuevos fármacos.

Otro ejemplo en el cual se utiliza la computación paralela es en la realización de tomografías médica por computador(TAC) que se usa mucho en los diagnósticos médicos y donde el tiempo necesario para realizar el diagnóstico es muy importante, en la figura 30 se muestra una imagen de tomografías médicas realizada por computadores y procesadas con clúster.

Render de imágenes: El Scientific Computing and Imaging y el Instituto de la Universidad de Utah ha explorado la visualización científica basado en clúster utilizando un clúster de visualización de 32 nodos compuesto por hardware comercial de componentes conectados con una red de alta velocidad. El OpenGL herramienta de visualización científica Simian se ha modificado para crear una versión compatible con clúster Simian que soporta paralelización haciendo uso explícito de los nodos del clúster remotos a través de una interfaz de paso de mensajes (MPI). Simian es capaz de generar imágenes 3D para simulaciones de incendios que se propagan de esa hipótesis de modelos tales como cuando un misil ubicado dentro de un grupo de combustible de avión se incendia y explota. Con la representación de imágenes para las simulaciones de incendios esta extensión permite a los investigadores tener una mejor visualización de los efectos destructivos.

Clasificación de clúster

Debido a que los clúster están formados por un grupo de computadoras y estas pueden tener diversas arquitecturas, existe la siguiente clasificación de acuerdo a su homogeneidad.

- Clúster homogéneos:

En este tipo de clúster todas las computadoras conectadas en red que forman los nodos del clúster tienen la misma arquitectura y utilizan un mismo sistema operativo (GNU/Linux, Unix o Windows), por ejemplo computadoras con arquitecturas Intel, AMD.

- Clúster heterogéneos:

En estos clúster todas las computadoras conectadas en red que forman los nodos del clúster tienen distintas arquitecturas y utilizan

sistemas operativos diferentes, como por ejemplo computadoras Intel, AMD y una multitud de diferentes sistemas operativos para los nodos. Controlar este entorno de un clúster heterogéneo es mucho más difícil que controlar un clúster homogéneo.

- **Meta clúster**

Estos son clúster de clúster, es decir los meta clúster son usualmente un grupo de clúster que están geográficamente distribuidos, ya sea en una nación o en el mundo entero, pero pueden ser tratados como un solo recurso de software especial y muy avanzado. Los meta clúster pueden ser también homogéneos o heterogéneos, los meta clúster también suelen ser llamados súper clúster.

Tipos de clúster

Según (Castillo Castillo, 2006)

Los clúster se pueden clasificar tomando en cuenta diferentes aspectos como la aplicación, disponibilidad, sistema operativo, configuración y el número de nodos.

Muchos de los trabajos de investigación han clasificado de manera muy similar los clúster, teniendo en cuenta ello para cuestiones prácticas se puede tomar los siguientes tipos de clúster.

- **Clúster de tolerancia a fallas(Fail-Over)**

Estos clúster utilizan una conexión de alto desempeño entre las computadoras, ésta conexión es para monitorear cual o cuales de los servicios están en uso, así como la sustitución de una máquina por otra cuando uno de sus servicios ha caído, por ejemplo los mecanismos de trabajo de Google Search Engine

- Clúster de Balanceo de carga (Load-Balancing)

Estos clúster son implementados cuando los recursos de los nodos son insuficientes para el procesamiento de datos, el clúster reparte toda la carga de trabajo entre los demás nodos del clúster.

- Alto rendimiento (High Performance)

Este tipo de clúster se implementa especialmente cuando centros de datos, centros de investigación entre otros necesitan y requieren gran capacidad de procesamiento y de tratamiento de datos así como una potencia de computación extrema.

Soluciones de clúster de computadoras de alto rendimiento

Durante la exploración tecnológica que se realizó se encontraron diversas soluciones para implementar clúster de computadoras de alto rendimiento que están basados e implementados sobre los clúster pioneros es decir Beowulf, Mosix y Openmosix, estos son: PelicanHPC, clusterknoppix, ABC Linux, Kerrighed, Rock Clúster y Quantian, todas estas soluciones son software libre ya que uno de los objetivos de la presente investigación es construir y realizar la investigación utilizando soluciones libres y todas estas soluciones están puestos en marcha sobre el sistema operativo GNU/Linux, se procede a continuación a realizar la descripción de cada una de estas soluciones así como de sus fundamentales características para posteriormente una comparativa entre ellos y elegir la plataforma para la investigación.

- Pelican HPC

PelicanHPC es una imagen ISO - híbrido (CD o USB) que permite configurar un clúster de computación de alto rendimiento en un par de minutos. Un clúster PelicanHPC permite hacer uso de la computación paralela MPI. Se puede ejecutar PelicanHPC en una sola máquina de núcleo múltiple de y así utilizar usar todos los

núcleos para resolver un problema en específico, o utilizar varios equipos de cómputo en red, La instalación del clúster Pelican es muy sencillo ya que se instala el nodo frontal o central y este a través de arranque de red proporciona los mecanismos para instalar los nodos desde una red local. Todos los nodos del clúster deben tener en su sistema de archivos la misma imagen, por lo que se garantiza que todos los nodos ejecutan el mismo software. Los paquetes instalados y configurados a través de APT. Este sistema operativo está basado en la distribución Debian. Puede utilizarse PelicanHPC orientado a diversas disciplinas, adicionalmente se puede crear una versión personalizada. No es muy difícil. Dos ejemplos de distribuciones especializadas que se suman a la base PelicanHPC son MOLA y birgHPC.

Estas son algunas características de Pelican HPC

- ✓ PelicanHPC está creado utilizando herramientas del proyecto Debian Live.
 - ✓ Contiene la última versión estable de la aplicación OpenMPI de MPI.
 - ✓ Disponible arquitecturas de 64 bits. también está disponible para 32 CPU bits.
 - ✓ Contiene programas de ejemplo utilizando GNU Octave. También tiene el punto de referencia Linpack HPL.
 - ✓ PelicanHPC y todas las pruebas se realiza mediante la versión estable de Squeeze versión de Debian GNU Linux como base.
- Clusterknoppix

Este sistema operativo basado en Knoppix y montado con Openmosix es un proyecto que surgió de la separación de los dos principales desarrolladores de Openmosix, es un sistema de clúster para el sistema operativo GNU/Linux que consiste en un parche aplicado en el

kernel responsable de las migraciones transparentes de procesos, y unas herramientas de área de usuario,

necesarias para calibrar y administrar el clúster. Esto permite que no se tenga que reprogramar nuestras aplicaciones para que aprovechen el clúster, es decir no hay que instalar la aplicación en todos los nodos para ejecutar una tarea.

- **ABCLinux**

La distribución ABC GNU/Linux está basada en Ubuntu y está especializada en la construcción automática de clúster Beowulf de alto rendimiento con tan solo arrancar el sistema en modo "live" en el frontend o siendo instalado en su disco duro. Los nodos arrancan a través de arranque por red. Usa como gestor de ventanas Gnome. Integra el monitor de recursos Ganglia. Se trata de la primera distribución que integra todas estas características. Ha sido publicado un artículo científico sobre este sistema en el IEEE y presentado en el Symposium ICAT2009 celebrado en Sarajevo (Bosnia & Herzegovina). Desarrollado por Iker Castaños Chavarri en el Departamento de Ingeniería de Sistemas y Automática de la EUITI de Bilbao, Universidad del País Vasco.

- **Kerrighed**

Kerrighed es una sola imagen del sistema del sistema operativo para clúster Kerrighed ofrece el punto de vista de una única máquina. Los objetivos de Kerrighed son la facilidad de uso, alto rendimiento de las aplicaciones, la alta disponibilidad del clúster, la gestión eficiente de los recursos, y capacidad de personalización de alta del sistema operativo.

- **Rock Clúster**

Rocks es una distribución de Linux basada en CentOS y un sistema de gestión de clúster que permite el despliegue rápido de clúster Linux

sobre hardware físico o contenedores virtuales Xen. Un clúster de Rocks, es fácil de implementar y ofrece todas las ventajas de la virtualización de los nodos del clúster. Con un mínimo de dos máquinas físicas, Rocks permite un despliegue y una gestión de clúster simple y rápida, liberando al administrador para que se centre en el apoyo a la red informática y a las aplicaciones distribuidas que hacen del clúster una opción atractiva. Rock clúster dispone de rollos para instalar herramientas así como software incluidas en la distribución estándar de Rocks, existen diferentes herramientas de código abierto de computación paralela y distribuida de alto rendimiento, tales como Sun Grid Engine , OpenMPI y Condor. Esta poderosa colección de funciones avanzadas es una de las razones por las que la NASA, la NSA, el Laboratorio de Investigación de IBM en Austin, la Marina de los EE.UU., el MIT, Harvard y la Universidad Johns Hopkins están todos utilizando Rocks por alguna de sus aplicaciones más intensivas.

- **Cuantian**

El sistema operativo Quantian es una remasterización de Knoppix / Debian para las ciencias computacionales. El medio ambiente es la a autoconfiguración y directamente de arranque de CD / DVD que convierte cualquier PC o portátil (siempre que se pueda arrancar desde CD-ROM / DVD) en un una estación de trabajos. Quantian también incorpora Clusterknoppix y añade soporte para Openmosix, incluyendo el inicio remoto de clientes ligeros en un contexto de servidor de terminal Openmosix permite una rápida configuración de un SMP clúster ordenador, se muestra en la figura 35 el funcionamiento de Quantian.

2.3. Definiciones conceptuales

- ✓ AMD: es una compañía estadounidense de semiconductores establecida en Sunnyvale, California, que desarrolla procesadores de cómputo y productos tecnológicos relacionados para el mercado de consumo
- ✓ CPU: La unidad central de procesamiento o unidad de procesamiento central (conocida por las siglas CPU, del inglés: central processing unit), es el hardware dentro de un ordenador u otros dispositivos programables, que interpreta las instrucciones de un programa informático mediante la realización de las operaciones básicas aritméticas, lógicas y de entrada/salida del sistema
- ✓ FRONTEND: Front-end y back-end son términos que se refieren a la separación de intereses entre una capa de presentación y una capa de acceso a datos, respectivamente
- ✓ KERNEL: En informática, un núcleo o kernel (de la raíz germánica Kern, núcleo, hueso) es un software que constituye una parte fundamental del sistema operativo, y se define como la parte que se ejecuta en modo privilegiado (conocido también como modo núcleo).
- ✓ MPI: Interfaz de Paso de Mensajes) es un estándar que define la sintaxis y la semántica de las funciones contenidas en una biblioteca de paso de mensajes diseñada para ser usada en programas que exploten la existencia de múltiples procesadores.
- ✓ OPENGL: es una especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D

2.4. Hipótesis

Hipótesis General

La utilización de un clúster de alto rendimiento virtual incrementará la velocidad de cálculo de programas.

Hipótesis Específicas

- A. La instalar y configuración del sistema operativo GNU/Linux con la distribución PelicanHPC permitirá la implementación del Clúster de alto rendimiento virtual
- B. La comprobación del incremento de velocidad de cálculo de los programas será satisfactoria mediante el uso del clúster de alto rendimiento virtual.

2.5. Variables

2.5.1. Variable Independiente

X: Clúster de alto rendimiento.

2.5.2. Variable Dependiente

Y: Caculo de los programas.

2.6. Operacionalización de Variables

VARIABLES	DIMENSIONES	INDICADORES	TIPO
Dependiente Cálculo de programas	Velocidad de Calculo	Tiempo de cálculo.	Numérica
Independiente Clúster de alto rendimiento	Clúster de alto rendimiento	Nodos Almacenamiento Sistemas operativos Conexiones de red Middleware Protocolos de comunicación y servicios Aplicaciones Ambientes de programación paralela	SI NO

CAPÍTULO III

METODOLOGÍA DE LA INVESTIGACIÓN

3.1. Tipo de Investigación

3.1.1. Enfoque

El presente estudio de investigación tiene el enfoque cuantitativo (Sampieri, 2006), “usa la recolección de datos para probar hipótesis, con base en la medición numérica y el análisis estadístico, para establecer patrones de comportamiento y probar teorías”. En este caso se realizan pruebas de rendimiento, midiendo la velocidad y tiempo de cálculo de la ejecución de algunos programas, el resultado son cifras numéricas que será procesadas y analizadas en forma numérica en el software estadístico.

3.1.2. Alcance

Esta investigación por su naturaleza es de nivel descriptivo y aplicativo, ya que utiliza la tecnología para demostrar que al utilizar un clúster de alto rendimiento se pueden incrementar la velocidad de cálculo de programas. Se describirá el proceso de implementación y de pruebas.

3.1.3. Diseño

El diseño que presenta el estudio de investigación es el cuasi experimental de pre y post prueba en el grupo de la investigación:

G: O1 X O2

Dónde:

G = Grupo de investigación (Calculo de programas)

X = Aplicación (Clúster de alto rendimiento)

O₁ = Pre Observación

O₂ = Post Observación

3.2. Población y Muestra

Por la naturaleza de la investigación no se cuenta con una población específica, debido a que la investigación se realiza en un escenario virtual, y solo tomando como unidad de análisis a la capacidad de cálculos de los programas, la muestra sería las cantidades de procesamiento de cálculo de los programas que se ejecutarán en base al clúster de alto rendimiento, en tal sentido por cada programa se realizarán 5 pruebas de las cuales se obtendrán resultados antes y después de la aplicación del clúster.

3.3. Técnicas e instrumentos de recolección de datos

Se utilizará la observación directa como técnica y la ficha de evaluación técnica como instrumento. En la ficha de evaluación técnica se anotarán los resultados por cada prueba ejecutada al momento de procesar los cálculos de cada programa.

3.4. Técnicas para el procesamiento y análisis de la información

Para el procesamiento de la información se usará la hoja de cálculo Excel para tabular los resultados de cada prueba y luego se analizarán en el programa SPSS procesando y mostrando la información en los respectivos organizadores de datos como son las tablas y los gráficos estadísticos.

CAPITULO IV

RESULTADOS

A continuación, se presenta el análisis de los datos, considerando que se trata de una variable numérica (tiempo en segundos).

4.1 PROCESAMIENTO DE DATOS

Cuadro N.º 1: Estadísticas de muestras relacionadas

		Media	N	Desviación estándar	Media de error estándar
r 1	Pa e	16,6384	5	,94827	,42408
	Po st	11,5844	5	1,41765	,63399

Fuente: Datos recopilados por el investigador

La tabla muestra las medidas de resumen y de dispersión necesarios para tener en cuenta las variaciones obtenidas. Se tiene que en promedio, se redujo en poco más de 5 segundos luego de la intervención.

4.2 CONTRASTACION DE HIPOTESIS Y PRUEBA DE HIPOTESIS

Ante todo, es necesario realizar la prueba de normalidad a los datos, en virtud de que se trata de datos numéricos, para determinar la pertinencia de uso de un procedimiento paramétrico o no paramétrico.

Cuadro N.º 2: Prueba de Kolmogorov-Smirnov para una muestra

		Pre	Post
N		5	5
Parámetros normales ^{a,b}	Media	16,638	11,584
		4	4
	Desviación estándar	,94827	1,4176
			5
Máximas diferencias extremas	Absoluta	,257	,191
	Positivo	,257	,191
	Negativo	-,161	-,173
Estadístico de prueba		,257	,191
Sig. asintótica (bilateral)		,200 ^{c,d}	,200 ^{c,d}

a. La distribución de prueba es normal.

b. Se calcula a partir de datos.

c. Corrección de significación de Lilliefors.

d. Esto es un límite inferior de la significación verdadera.

Fuente: resultados del procesamiento en el software

De la tabla anterior, deducimos que los datos tienen distribución normal, debido a que el nivel de significancia o p-valor es mayor a 0.05, por lo tanto, esto nos indica que es pertinente el uso de un procedimiento estadístico no paramétrico para el tratamiento de los datos. En el presente estudio, la comparación de medias se realiza utilizando t de Student para muestras relacionadas, con cada uno los tiempos.

Prueba de hipótesis Comparación antes-después, evaluación de tiempos (segundos)

Se tiene la información de que, bajo el sistema clásico (sin el sistema), el tiempo promedio del pretest es de 16,6384 segundos. Una vez que se opera con el nuevo sistema se ha evaluado nuevamente, obteniéndose un promedio de tiempo de 11,5844 segundos. Se desea constatar si la utilización de un clúster de alto rendimiento virtual incrementa la velocidad de cálculo de programas.

Cuadro N.º 3: El ritual de la significancia estadística

1	Plantear Hipótesis Ho: La utilización de un clúster de alto rendimiento virtual incrementa la velocidad de cálculo de programas. H1:
2	Establecer un nivel de significancia Nivel de Significancia (alfa) $\alpha = 5\% = 0.05$
3	Seleccionar estadístico de prueba: Prueba t de Student para muestras relacionadas
4	Valor de P= 0,006982 = Lectura del p-valor: Con una probabilidad de error del 0.6982% la utilización de un clúster de alto rendimiento virtual incrementa la velocidad de cálculo de programas.
5	Toma de decisiones Con una probabilidad de error del 0.6982% la utilización de un clúster de alto rendimiento virtual incrementa la velocidad de cálculo de programas.

Fuente: resultados del procesamiento en el software

Interpretación

Se deduce que la utilización de dicho clúster efectivamente incrementa la velocidad de cálculo de programas.

CAPÍTULO V

DISCUSIÓN DE RESULTADOS

En el capítulo a continuación se da a conocer los resultados de la investigación aplicada denominada: Computación Paralela: Clúster de Alto Rendimiento Virtual para el incremento de velocidad de Cálculo de Programas.

El experimento fue llevado a cabo en un entorno virtual usando la tecnología de la virtualización de sistemas y entornos computacionales, en ese sentido se llegó a virtualizar, dos nodos, dos computadores bajo el sistema operativo GNU/Linux para demostrar el trabajo paralelo de estos dos nodos, bajo la metodología clustering.

En la tabla 4.1 podemos observar una diferencia de medias entre el antes y después de la aplicación, vemos que el pre se obtiene una media de 16,63 segundos y en el post una media de 11,58, esto se explica que hubo una reducción de 5 segundos al momento de ejecutar la aplicación para realizar el cálculo previsto usando dos nodos.

Cuadro N° 4: Resultados del número de repeticiones de las pruebas con uno y dos nodos

Nº repeticiones	pre	post
	1 nodo	2 nodos
tiempo seg.		
1	15.701	13.809
2	16.058	12.025
3	18.052	10.25
4	17.125	11.256
5	16.256	10.58

Fuente: resultados del procesamiento en el software

Como podemos observar se realizaron 5 repeticiones del programa para realizar un cálculo, con un solo nodo y posteriormente con dos nodos.

Se obtiene un promedio de 11,5844 segundos en el post test en comparación con el pre test de 16,6384 segundos, con una diferencia de 5.054 segundos, esto es: que usando dos nodos la velocidad de cálculo del programa se reduce a 5 segundos, y con esto demostramos que en una malla o clúster de computadoras se optimiza o se incrementa la velocidad de cálculo de los programas.

CONCLUSIONES

- ✓ Se instaló y se configuró el sistema operativo GNU/Linux con la distribución PelicanHPC para la implementación del Clúster de alto rendimiento virtual; para esto se usaron máquinas virtuales para la aplicación y para las pruebas necesarias.
- ✓ Se incrementó la velocidad de cálculos de los programas al momento de usar dos nodos, utilizando en forma paralela el poder de procesamiento de las máquinas virtuales.
- ✓ Para las pruebas necesarias se usó un programa de computadora de cálculo básico, programado y orientado a la utilización de los nodos en forma simultánea.

RECOMENDACIONES

- ✓ Para investigaciones posteriores, se recomienda usar el clúster de alto rendimiento con máquinas físicas que tengan la misma arquitectura de hardware, mejor dicho, con la misma capacidad de procesamiento y memoria, en otras palabras, para la implementación de un clúster homogéneo.
- ✓ La implementación de un clúster se puede emplear para la optimización de procesos y cálculo en el área de gráficos o de cálculos científicos avanzados, en tal situación se recomienda usar más de 4 nodos que puedan potenciar y mejorar la velocidad de cálculo y procesamiento.
- ✓ Ante de usar el sistema operativo para la ejecución del clúster se recomienda realizar las actualizaciones necesarias del sistema, y realizar las instalaciones correspondientes de los archivos necesarios para el funcionamiento del clúster.

REFERENCIAS BIBLIOGRÁFICAS

- Arrollo, R., Nievas, F., & Pino, O. (. (2004). *Los clusters como Plataforma de Procesamiento Paralelo*. Obtenido de http://usuarios.lycos.es/lacaraoculta/descargas/Clusters_definitivo.pdf
- Branch Bedoya, J. W., & Mesa Munera, A. (2008). Implementación de un cluster homogéneo para la solución de problemas de alta complejidad computacional. *Avances en Sistemas e Informática*, 5.
- Branch Bedoya, J. W., & Mesa Munera, A. (2008). *Implementación de un cluster homogéneo para la solución de problemas de alta complejidad computacional*.
- BULEJE, C. Y. (2014). *INFLUENCIA DE UN CLÚSTER DE COMPUTADORAS DE ALTO RENDIMIENTO EN EL TIEMPO DE RENDERIZACIÓN DE MODELOS 3D FOTOREALISTAS, UNIVERSIDAD NACIONAL JOSÉ MARÍA ARGUEDAS 2013. ANDAHUAYLAS.*
- CÁCERES, G. (2012). *ESTRATEGIA DE IMPLEMENTACIÓN DE UN CLÚSTER DE ALTA DISPONIBILIDAD DE N NODOS SOBRE LINUX USANDO SOFTWARE LIBRE*. Quito.
- Castillo Castillo, J. &. (2006). *Implementación de un Cluster Openmosix para Cómputo Científico*. Mexico.
- Castillo Castillo, J., & Olivera Acosta, R. B. (. (2006). *Implementación de un Cluster Openmosix para Cómputo Científico*. Mexico.
- Castillo Castillo, J., & Olivera Acosta, R. B. (2006). *Implementación de un Cluster Openmosix para Cómputo Científico*. México.
- Chirinov, R. (. (4 de marzo de 2004). *Proyecto Cluster Openmosix(Linux)*. Obtenido de <http://www.noticias3d.com/articulo.asp?idarticulo=248pag=4>

- Correa, M. (2 de enero de 1999). *Instalación y Uso del Cluster Beowulf en Ceca/CULA*. Obtenido de <http://www.cecalc.ula.ve/documentacion/tutoriales/beowulf/node1.html>
- Dormido, S., Hernández, R., Ros, S., & Sánchez, J. (2003). *Procesamiento Paralelo Teoría y Programación*. Madrid.
- GONZÁLEZ RAMÍREZ EDGAR RUBÉN y RODRÍGUEZ SÁNCHEZ ABIMAE. (2008). *DISEÑO E IMPLEMENTACIÓN DE UN CLUSTER TIPO BEOWULF PARA EL DESARROLLO DE CÓMPUTO CIENTÍFICO AVANZADO*. México.
- Llorens, E., & Peña, M. (s.f.). *Computación Cluster Beowulf*. Obtenido de <http://personals.ac.upc.edu/enric/PFC/Beowulf/beowulf.html>
- Martínez, F. O. (2004). *IMPLEMENTACIÓN DE UN SERVIDOR WEB VIRTUAL BALANCEADOR DE CARGA BASADO EN LINUX*. Colima.
- Pérez, M. (2001). *Arquitecturas paralelas*. Obtenido de <http://www.dragones.org/Biblioteca/Articulos/ArquitecturaParalela2.pdf>
- Robbins, D. (. (s.f.). *Openmosix*. Obtenido de <http://www.intel.com/cd/ids/developer/asmo%ADna/eng/20449.htm>
- Sampieri, R. (2006). *Metodología de la investigación*. Mexico: Mc Graw Hill.
- Sánchez, E., & Heider, Y. (2007). *Cluster and Grid Computing*. Mexico: Prentice Hall.
- Sterling, T. S. (1999). *How to Build a Beoful: a guide to the implementation and aplicacion of PC cluster*. Cambridge.
- Sun. (22 de Enero de 2006). *Sun Cluster Architecture*. Obtenido de Sun Cluster Architecture: <http://www.sun.com./software/grid/SunCLusterArchitecture.pdf>
- Torrealba Martinez, L. M. (2002). *Construcción de un Cluster Mosix con Pruebas con Simulación de Halo*. Oxaca, Huajuapan. Mexico.
- Zamora, L. J. (2014). *CLÚSTER DE SERVIDORES LINUX PARA ALTA DISPONIBILIDAD DE LA INFORMACION*. Cajamarca,.

ANEXOS

ANEXO 01: MATRIZ DE CONSISTENCIA

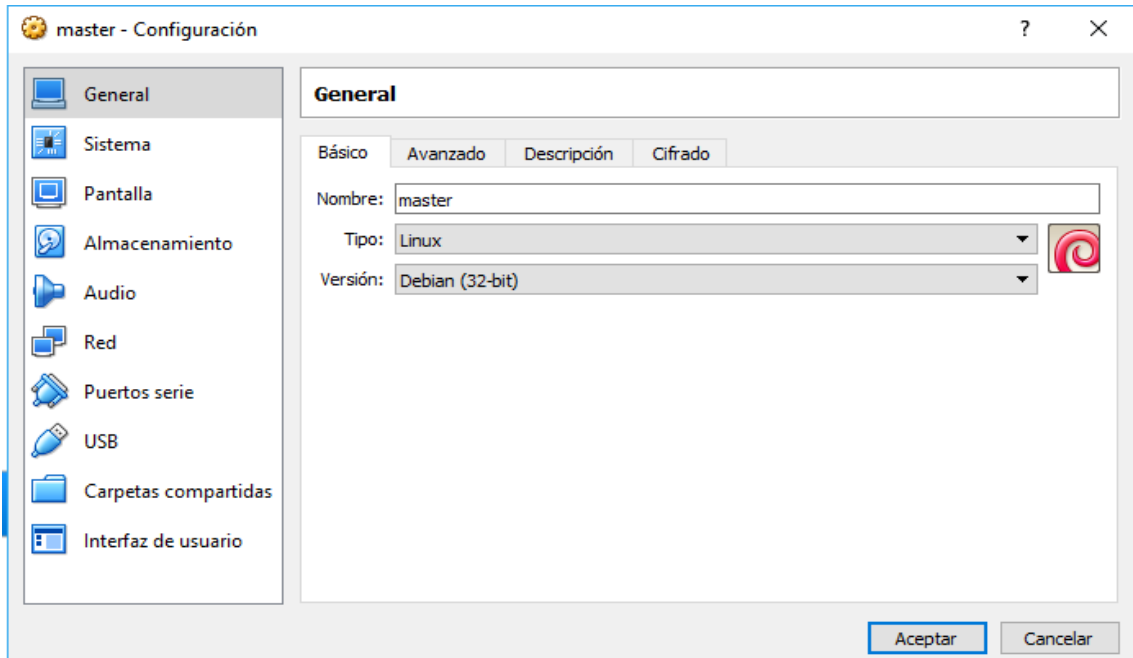
PROBLEMAS	OBJETIVOS	HIPÓTESIS	VARIABLES	DIMENSIONES	INDICADORES	METODOLOGÍA
<p>Problema General</p> <p>¿En qué medida se incrementa la velocidad de cálculo de programas empleando un clúster de alto rendimiento virtual?</p>	<p>Objetivo General</p> <p>Demostrar que con la utilización del clúster de alto rendimiento virtual se incrementa la velocidad de cálculo de programas.</p>	<p>Hipótesis General</p> <p>La utilización de un clúster de alto rendimiento virtual incrementará la velocidad de cálculo de programas.</p>	<p>Dependiente</p> <p>Cálculo de programas</p>	<p>Velocidad de Calculo</p>	<p>Tiempo de cálculo.</p>	<p>Enfoque: Cuantitativo</p> <p>Tipo: Descriptivo - Aplicativo</p> <p>Diseño: Pre-experimental</p>
<p>Problema Específico</p> <p>P.E 01: ¿Cuál es el procedimiento para Instalar y configurar el sistema operativo GNU/Linux con la distribución PelicanHPC para la implementación del Clúster de alto rendimiento virtual?</p> <p>P.E 02: ¿De qué forma se realiza la comprobación del incremento de velocidad de cálculos los programas mediante el uso del clúster de alto rendimiento virtual??</p>	<p>Objetivos Específico</p> <p>O.E.1: Instalar y configurar el sistema operativo GNU/Linux con la distribución PelicanHPC para la implementación del Clúster de alto rendimiento virtual</p> <p>O.E.2: Realizar la comprobación del incremento de velocidad de cálculo de los programas mediante el uso del clúster de alto rendimiento virtual.</p>	<p>Hipótesis Especifica</p> <p>H1: La instalación y configuración del sistema operativo GNU/Linux con la distribución PelicanHPC permitirá la implementación del Clúster de alto rendimiento virtual.</p> <p>H2: La comprobación del incremento de velocidad de cálculo de los programas será satisfactoria mediante el uso del clúster de alto rendimiento virtual.</p>	<p>Independiente</p> <p>Clúster de alto rendimiento</p>	<p>Clúster de alto rendimiento</p>	<p>Nodos Almacenamiento Sistemas operativos Conexiones de red Middleware Protocolos de comunicación y servicios Aplicaciones Ambientes de programación paralela</p>	<p>Esquema del Diseño:</p> <p>G: O1 X O2</p> <p>•Donde:</p> <p>G= Grupo de investigación (Calculo de programas)</p> <p>X= Aplicación de la variable</p> <p>O1, O2, = Medición de Observación</p>

TÍTULO DEL PROYECTO: COMPUTACIÓN PARALELA: CLUSTER DE ALTO RENDIMIENTO VIRTUAL PARA EL INCREMENTO DE VELOCIDAD DE CALCULO DE PROGRAMAS

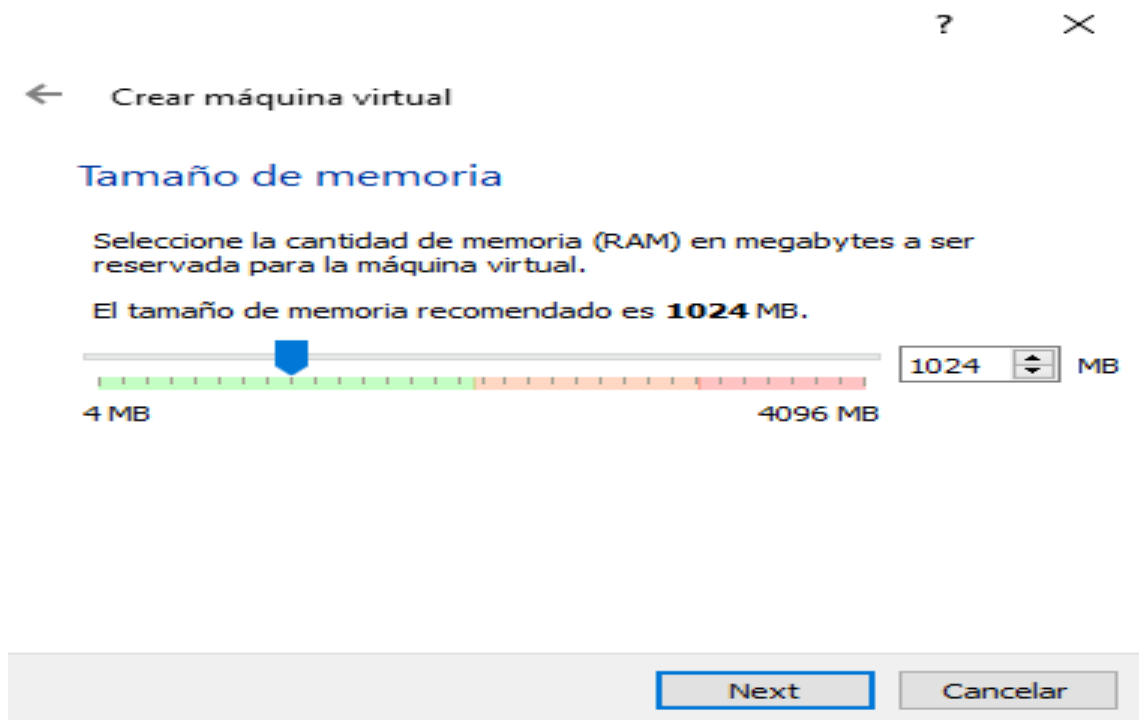
Creando Clúster con PelicaHPC LINUX EN MAQUINAS VIRTUALES

1. CREACIÓN DEL MASTER O SERVIDOR

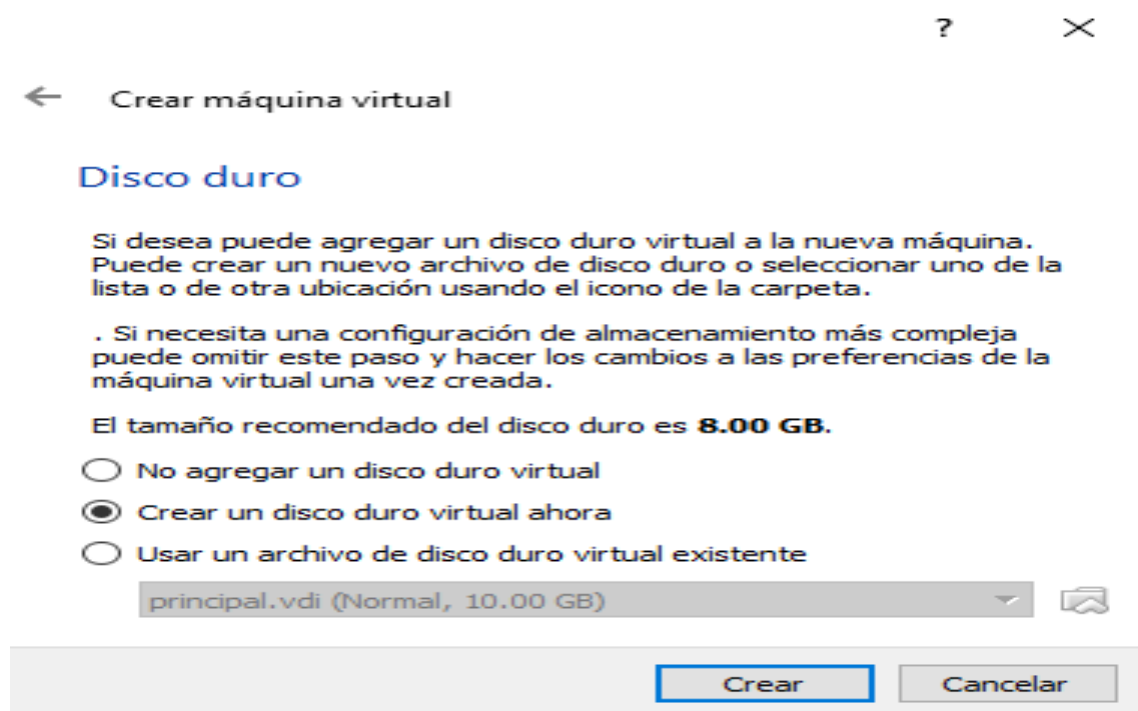
Pondremos un nombre cualquiera, en tipo ponemos linux y version ubuntu



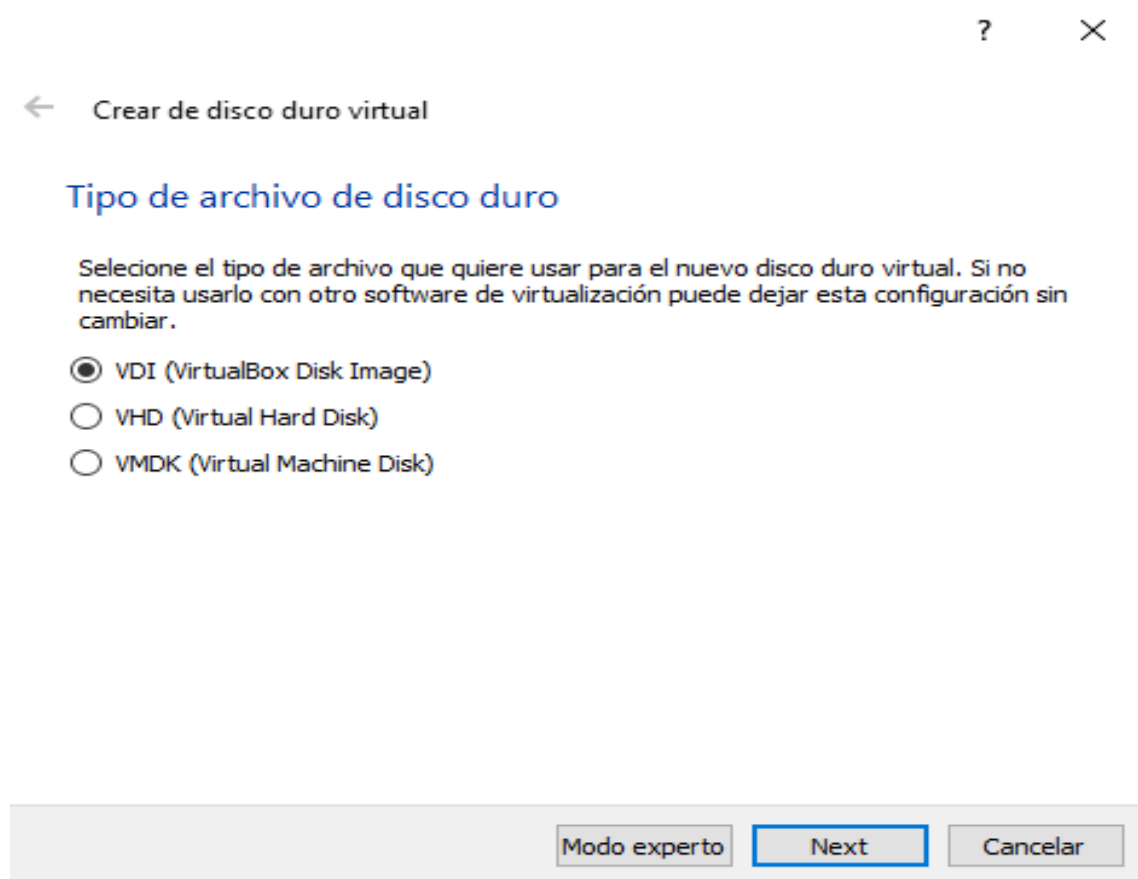
Tamaño de la memoria 1GB



Disco duro, lo dejamos como esta



Tipo de archivo de disco duro lo dejamos como esta



Almacenamiento en unidad de disco duro física

?

×

← Crear de disco duro virtual

Almacenamiento en unidad de disco duro física

Seleccione si el nuevo archivo de unidad de disco duro virtual debería crecer según se use (reserva dinámica) o si debería ser creado con su tamaño máximo (tamaño fijo).

Un archivo de disco duro **reservado dinámicamente** solo usará espacio en su disco físico a medida que se llena (hasta un máximo **tamaño fijo**), sin embargo no se reducirá de nuevo automáticamente cuando el espacio en él se libere.

Un archivo de disco duro de **tamaño fijo** puede tomar más tiempo para su creación en algunos sistemas, pero normalmente es más rápido al usarlo.

Reservado dinámicamente

Tamaño fijo

Next Cancelar

Ubicación del archivo y tamaño 8 GB


?

×


← Crear de disco duro virtual

Ubicación del archivo y tamaño

Escriba el nombre del archivo de unidad de disco duro virtual en el campo debajo o haga clic en el icono de carpeta para seleccionar una carpeta diferente donde crear el archivo.

master 

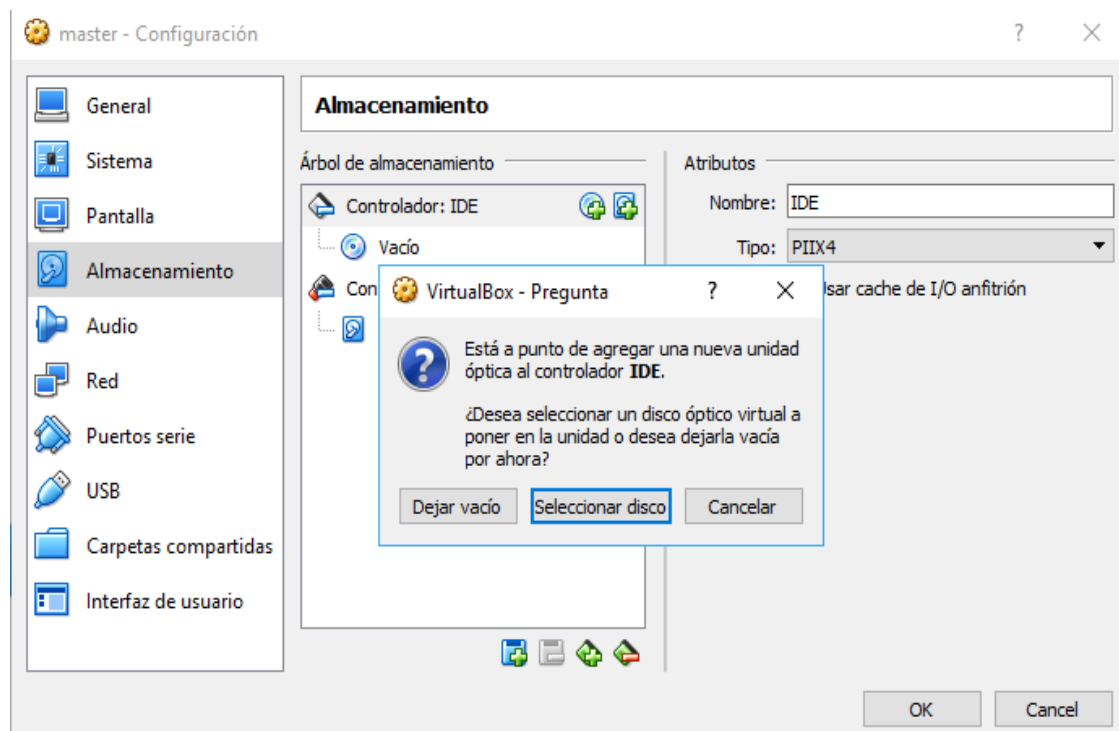
Seleccione el tamaño de disco duro virtual en megabytes. Este tamaño es el límite para el archivo de datos que una máquina virtual podrá almacenar en el disco duro.

4.00 MB  2.00 TB

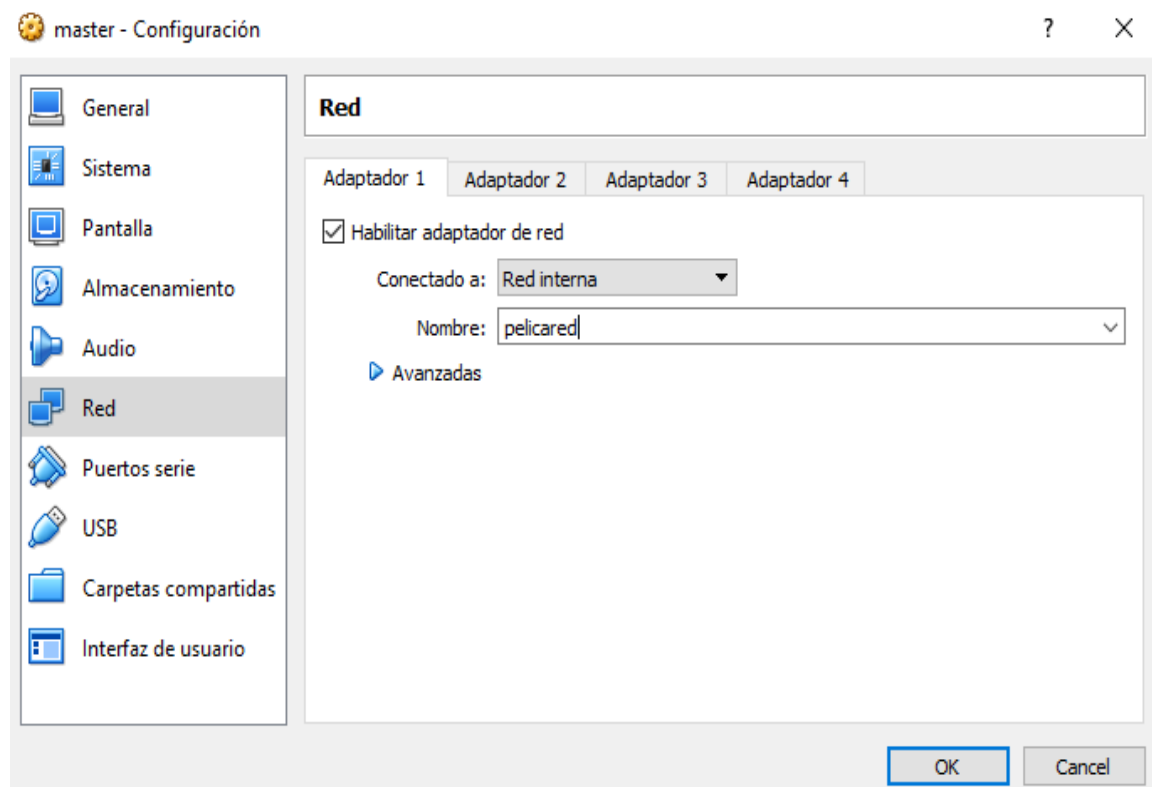
8.00 GB

Crear Cancelar

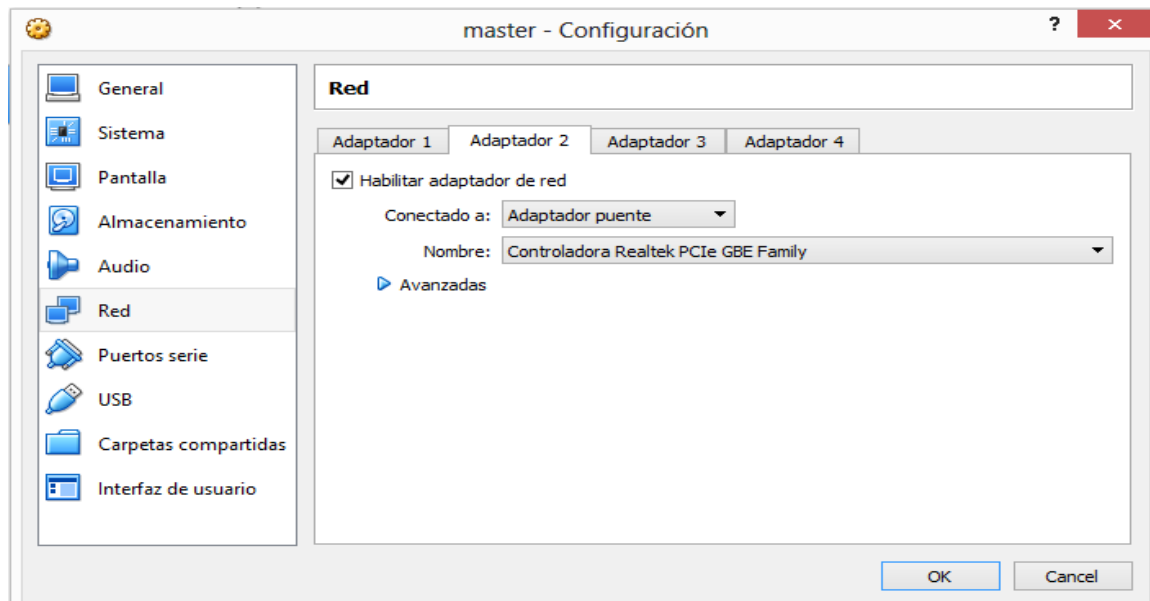
Cargamos iso configuración->almacenamiento



Configuración de la red interna ->adaptador 1(red interna y poner un nombre si lo desean)



Configuración de red interna -> Adaptador 2(red NAT o Adaptador Puente), no es necesario esta configuración solo s realiza para que el clúster servidor tenga internet



2. INICIAMOS INSTALACIÓN

Arrancamos en frontend con el Live CD de PelicanHPC. Escogemos la primera opción: Live.



PelicanHPC is a tool for the creation of a cluster for parallel computing using MPI. It is based on Debian GNU/Linux, using live-builder from the Debian Live project.

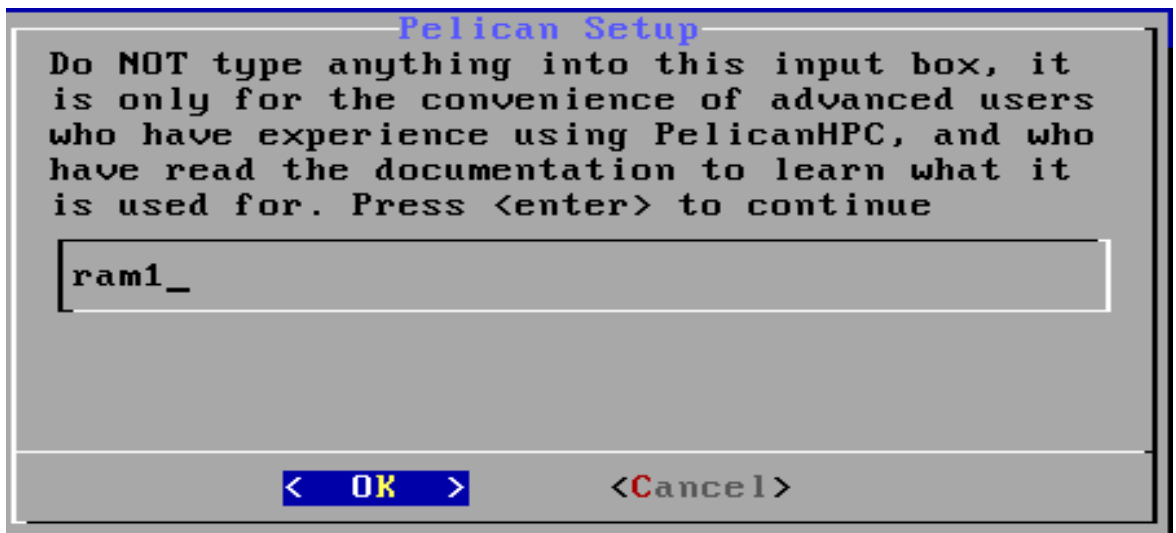
This is version
2.2. Special
thanks to
Robert G. Petry

Boot menu

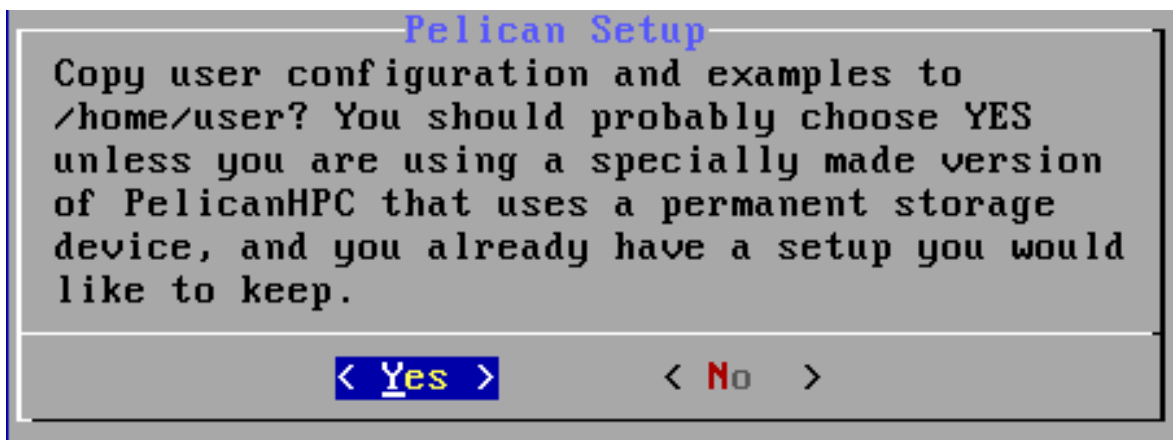
```
Live
Live (failsafe)
Memory test
Help
```

Automatic boot in 11 seconds...
Press ENTER to boot or TAB to edit a menu entry

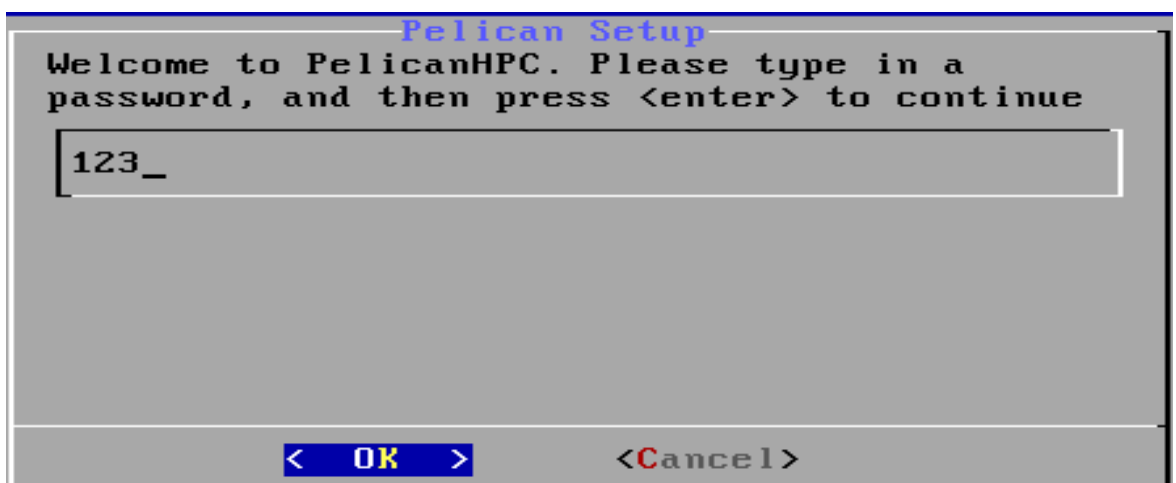
En esta ventanita le vamos OK



En esta ventana también le damos en OK



Escogemos un password



El siguiente paso es escribir el usuario **user** y la contraseña que pusimos anteriormente para la aplicación. Mediante esta contraseña los demás nodos se pueden conectar con el frontend.

```
Welcome to PelicanHPC!

To log in, enter user as the username, and the password you just specified.
After you're logged in, you can:

* create a cluster: type pelican_setup
* enter a desktop environment: type startx

For more information, visit http://pelicanhpc.org. Have fun!

Debian GNU/Linux 5.0 pel1 tty1

pel1 login: user
Password: _
```

Para seguir con la instalación colocamos el comando `pelican_setup` y luego presionamos Enter.

```
pel1 login: user
Password:
user@pel1:~$ pelican_setup_
```

Continúa la instalación. Si tenemos más de un dispositivo de red, debemos escoger con el que vamos a configurar el clúster.

En esta pantalla debemos seleccionar Yes, para iniciar los servicios. Debemos tener cuidado de que la red del clúster esté aislada de otras redes.

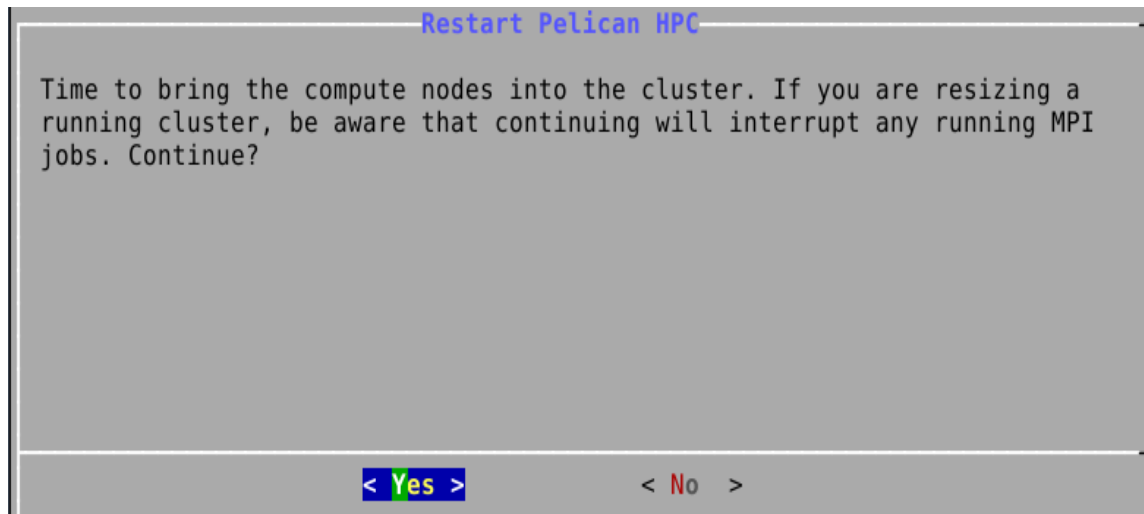
```
Start Pelican HPC netboot services
We now get ready to set up the cluster by starting services that will
allow the compute nodes to netboot. IMPORTANT: do not proceed if your
cluster is on an existing network, or PelicanHPC's dhcp server may
conflict with a running dhcp server. Continue?

.

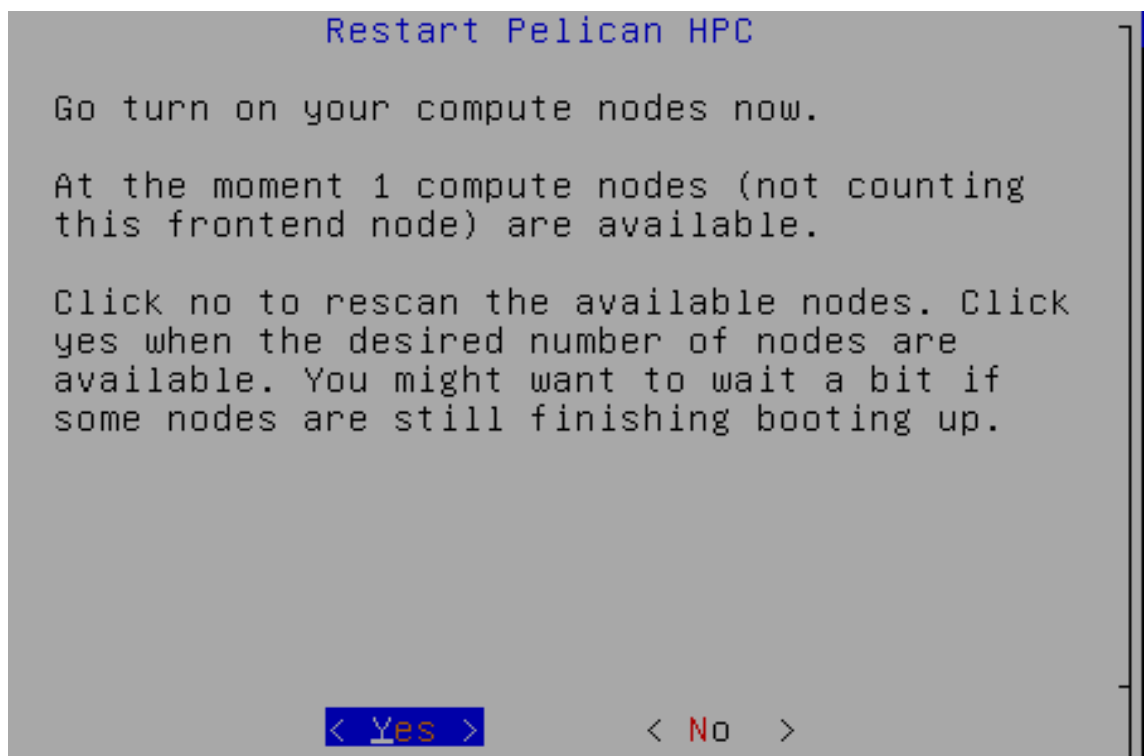
< Yes > < No >
```

En este punto debemos seleccionar Yes. También es momento de arrancar con el nodo.

Arrancamos el nodo con booteo desde la LAN. Comienza la instalación en el nodo.



Luego de pocos minutos el nodo del clúster está listo.



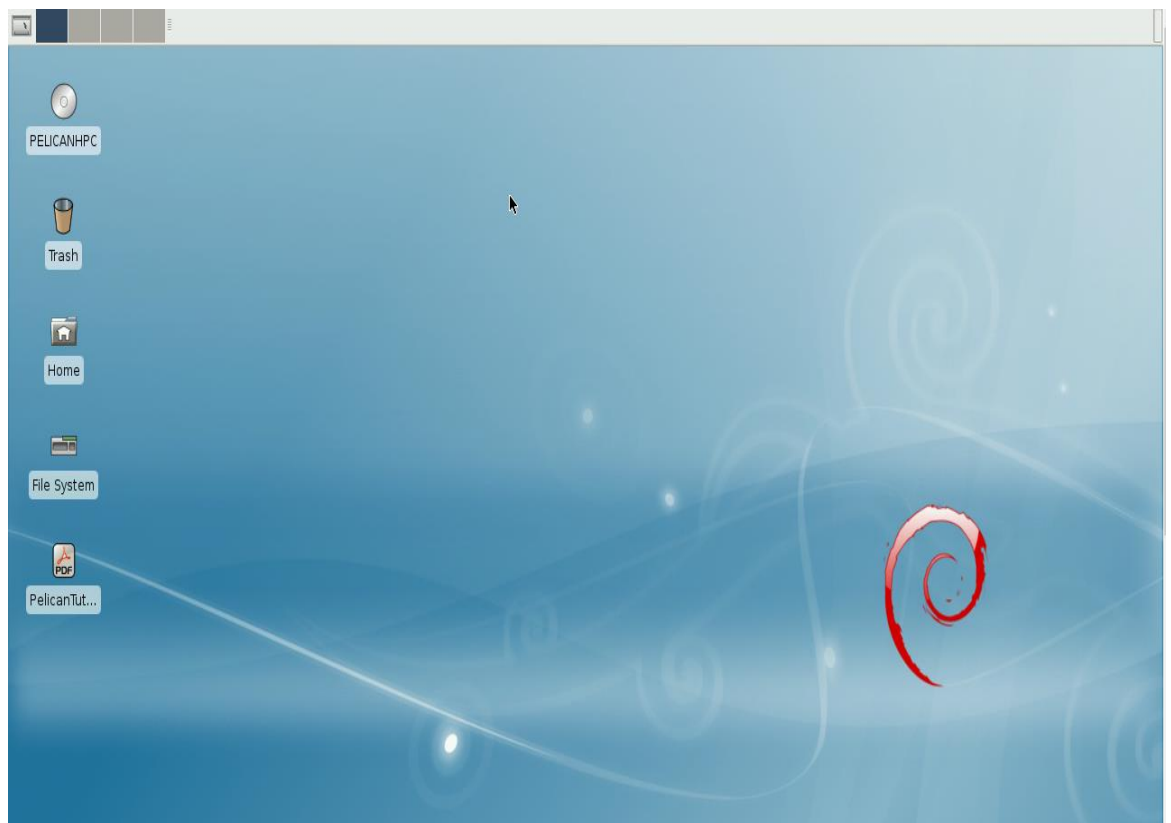
Para configurar el clúster en modo gráfico, venimos desde el paso 9. En este paso escribimos startx y Enter.

```
HPC Test -----
Quantity of processors = 2
Calculation time      = 0.88 seconds
Cluster speed         = 2039 MFLOPS
-----
Cluster node N00 speed = 1019 MFLOPS
Cluster node N01 speed = 1019 MFLOPS
-----

user@pel1:~$ _
```

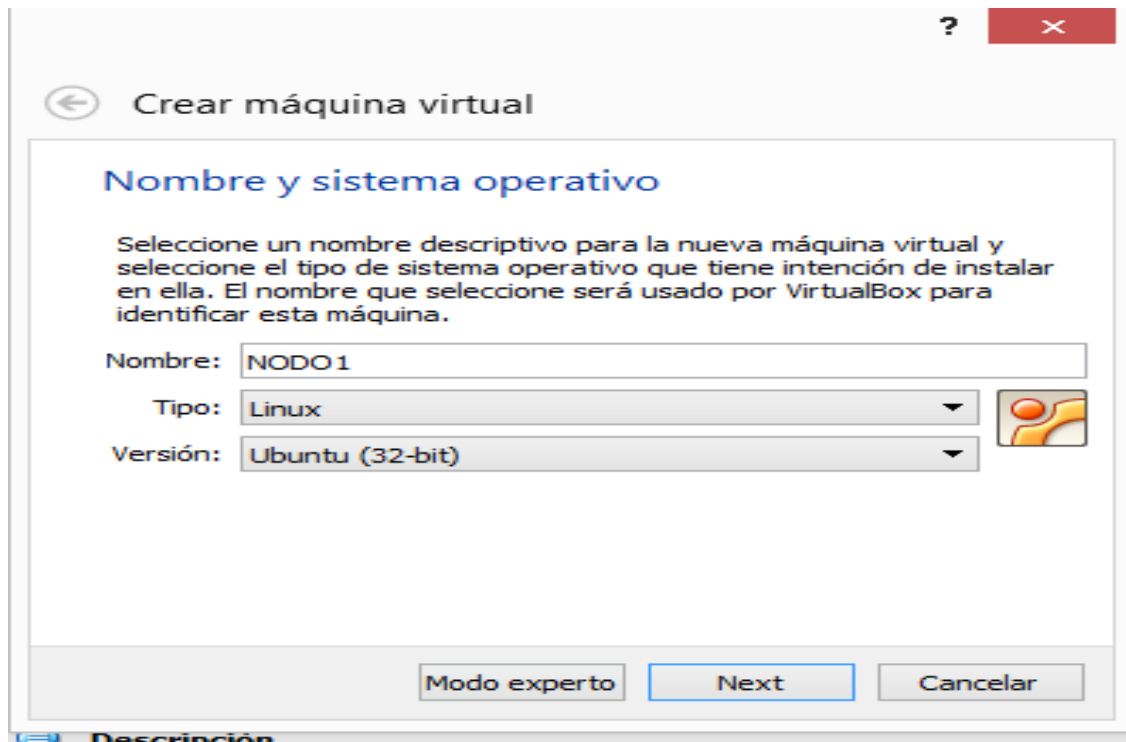
```
-----
user@pel1:~$ startx_
```

Nos aparecerá la siguiente pantalla. Aquí ponemos cualquiera de los 2 no afecta el Clúster

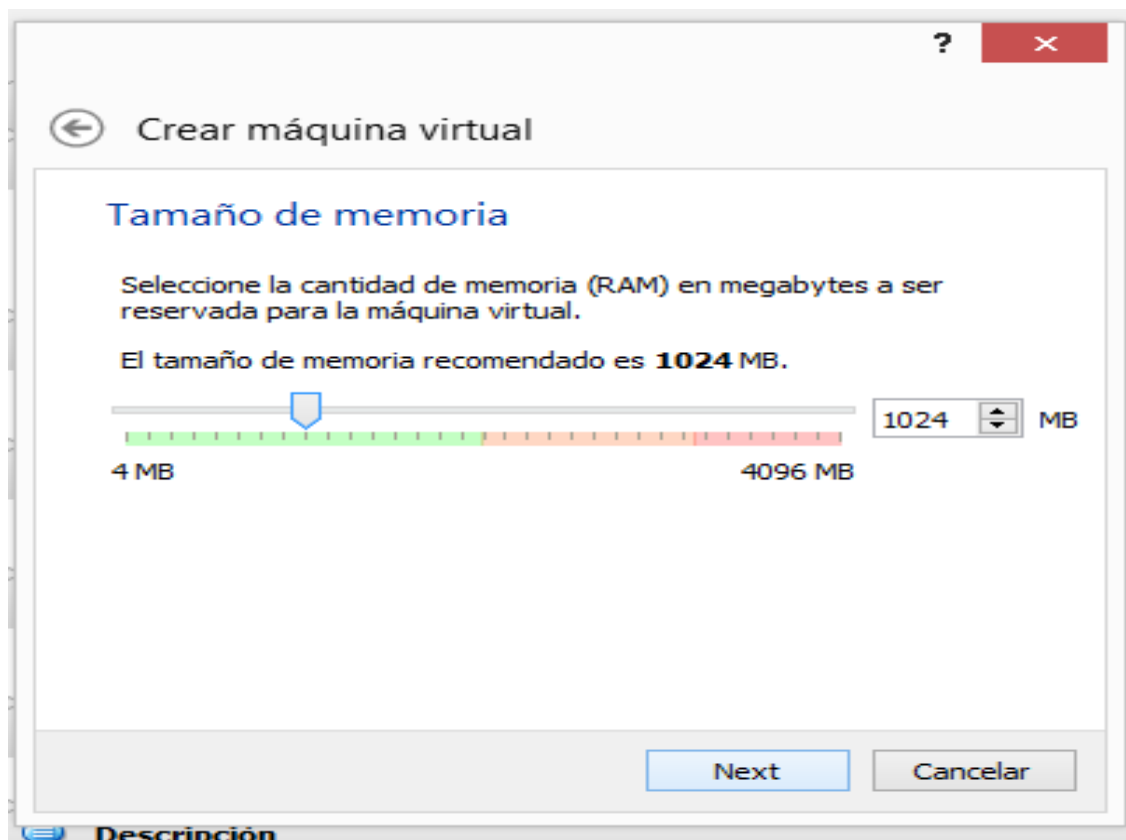


1. CREACION DE LOS NODOS A y B

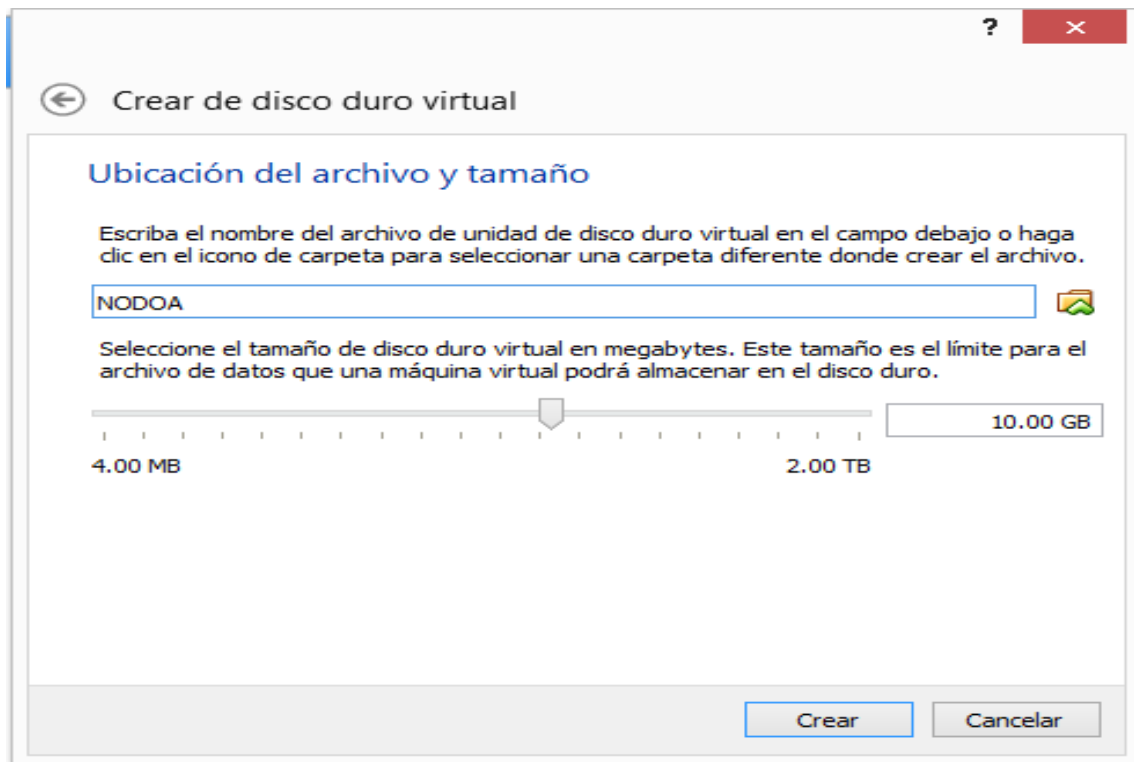
Nombre ponemos nodo1, tipo linux y version ubuntu



Tamaño de la memoria 1 GB

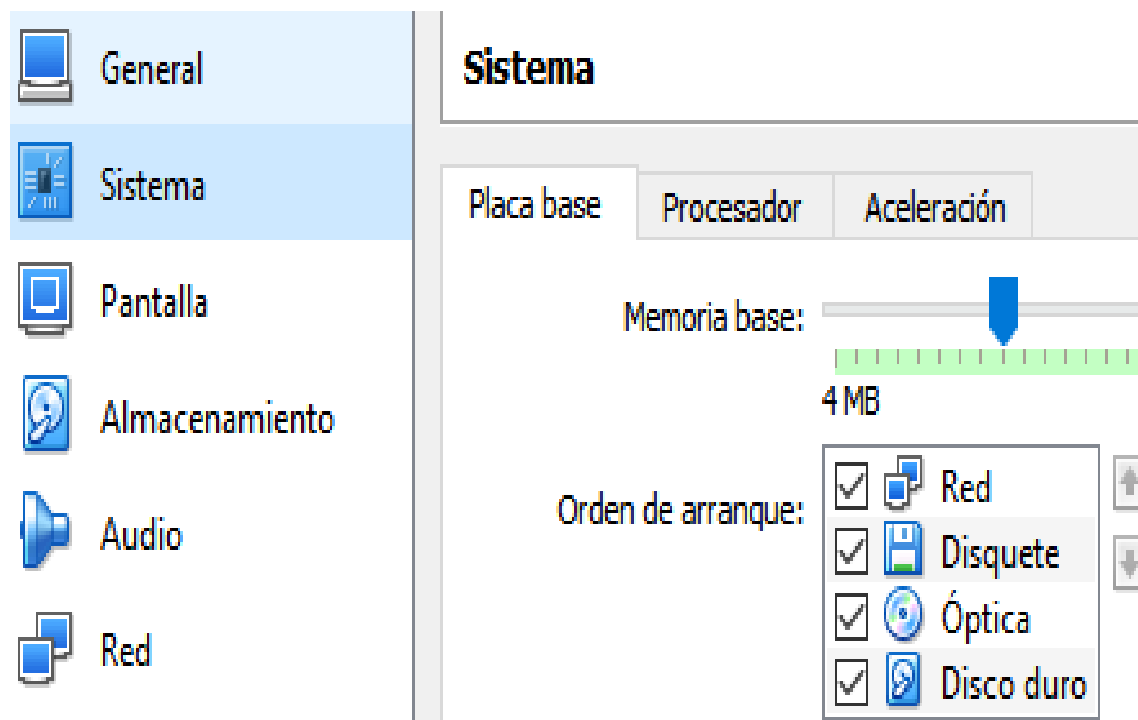


PROCESADOR DE 10 GB



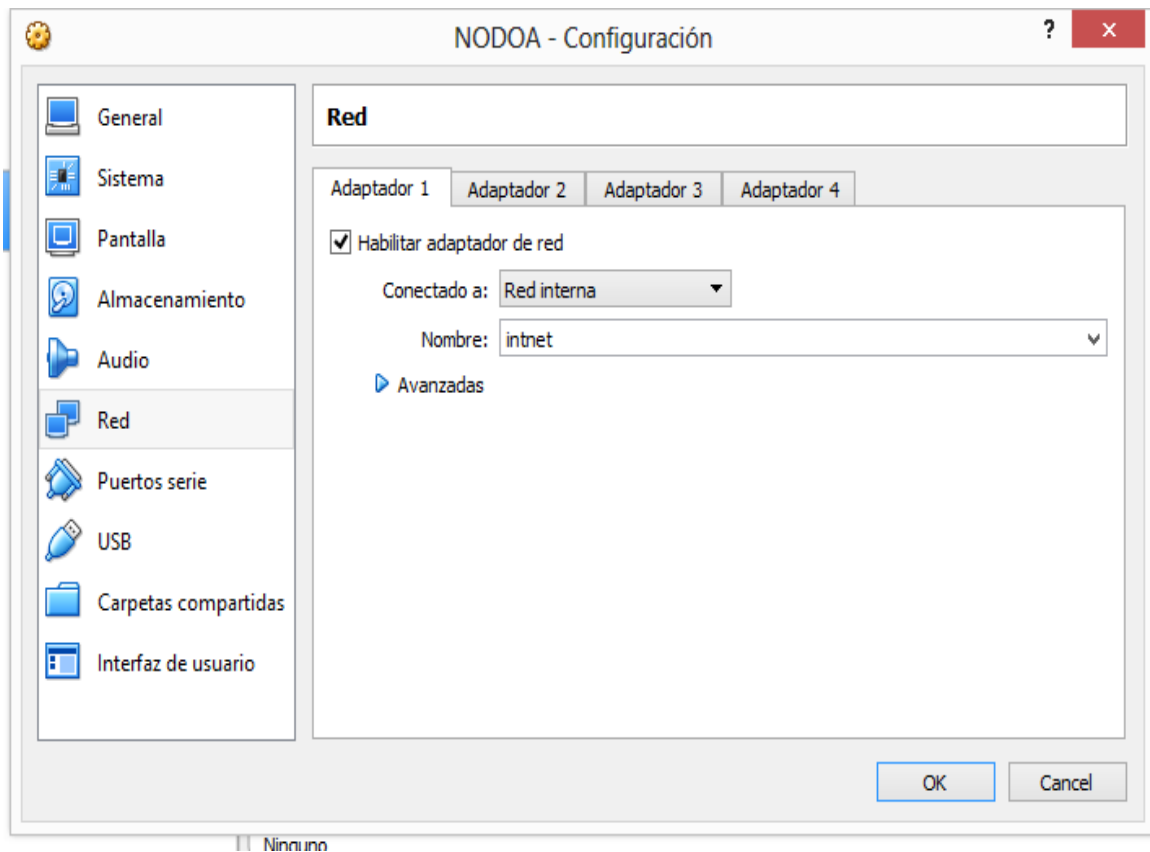
NODO A ->configuración->sistema

En orden de arranque ponemos desde la RED



Configuración de red->adaptador1

Red interna, ponemos un nombre cualquiera



Iniciamos nodo a para q se vincule al master

```
This is a PelicanHPC compute node. It is part of a cluster of computers that is
doing some REALLY important stuff.

Please don't try to use it, and DON'T TURN IT OFF!

THANKS!

Debian GNU/Linux 5.0 pel71 tty1

pel71 login: _
```

NOTA: hacemos los mismos procedimientos con el nodo B

Materiales:

Iso: pelicanhpc-v2.2+eps-i386

IP: 10.11.12.1

Comprobando rendimiento SUPER PI

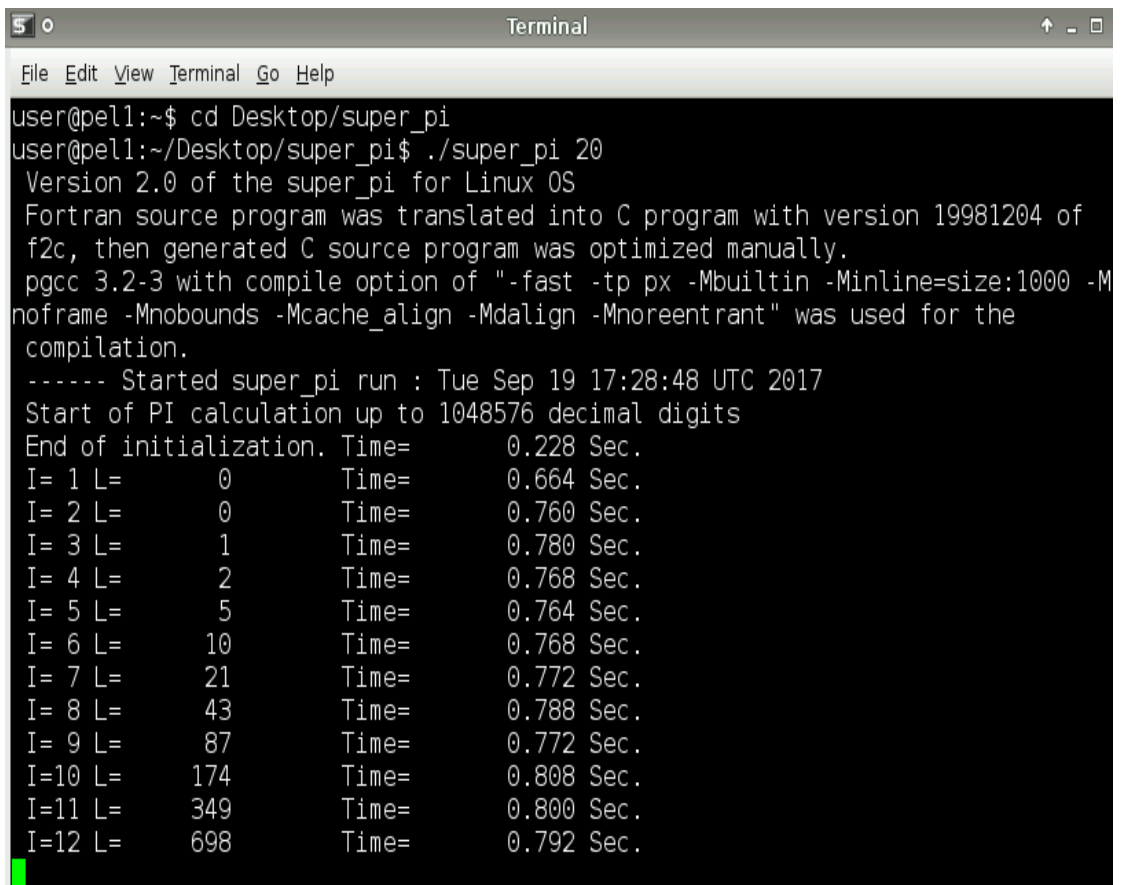
Super PI es un programa de computadora que calcula pi a un número especificado de dígitos después del punto decimal, hasta un máximo de 32 millones.

Instalación del súper PI 1.8

- ❖ Descomprimos el archivo super_pi.tar
- ❖ En terminal vamos a la ubicación del archivo y ponemos el siguiente comando y ponemos un valor en mi caso lo pondré 20

```
user@pell:~$ cd Desktop/super_pi
user@pell:~/Desktop/super_pi$ ./super_pi 20
```

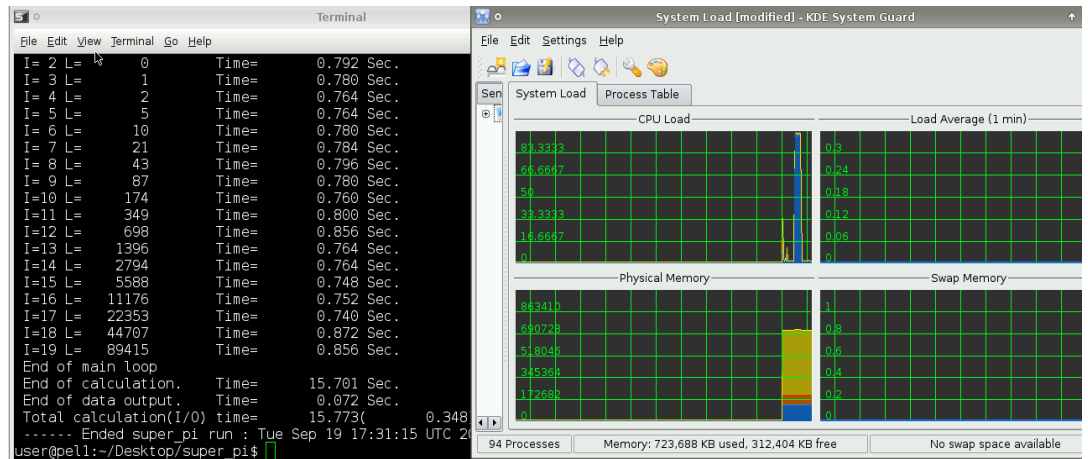
- ❖ Presionamos enter y nos saldrá algo así al terminar



```
Terminal
File Edit View Terminal Go Help
user@pell:~$ cd Desktop/super_pi
user@pell:~/Desktop/super_pi$ ./super_pi 20
Version 2.0 of the super_pi for Linux OS
Fortran source program was translated into C program with version 19981204 of
f2c, then generated C source program was optimized manually.
pgcc 3.2-3 with compile option of "-fast -tp px -Mbuiltin -Minline=size:1000 -M
noframe -Mnobounds -Mcache_align -Mdalign -Mnoreentrant" was used for the
compilation.
----- Started super_pi run : Tue Sep 19 17:28:48 UTC 2017
Start of PI calculation up to 1048576 decimal digits
End of initialization. Time=      0.228 Sec.
I= 1 L=      0      Time=      0.664 Sec.
I= 2 L=      0      Time=      0.760 Sec.
I= 3 L=      1      Time=      0.780 Sec.
I= 4 L=      2      Time=      0.768 Sec.
I= 5 L=      5      Time=      0.764 Sec.
I= 6 L=     10      Time=      0.768 Sec.
I= 7 L=     21      Time=      0.772 Sec.
I= 8 L=     43      Time=      0.788 Sec.
I= 9 L=     87      Time=      0.772 Sec.
I=10 L=    174      Time=      0.808 Sec.
I=11 L=    349      Time=      0.800 Sec.
I=12 L=    698      Time=      0.792 Sec.
```

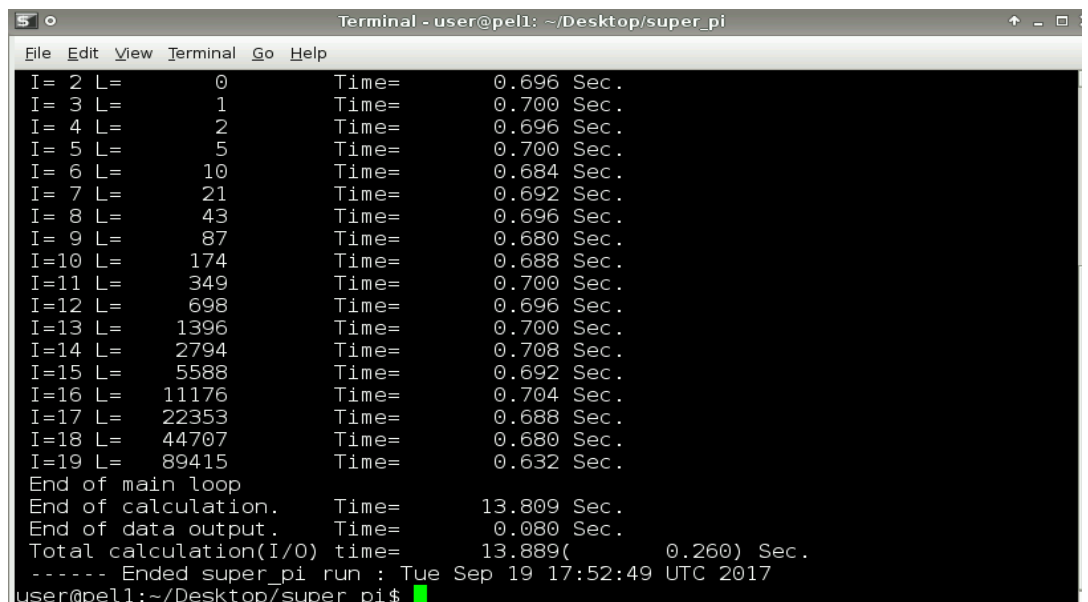
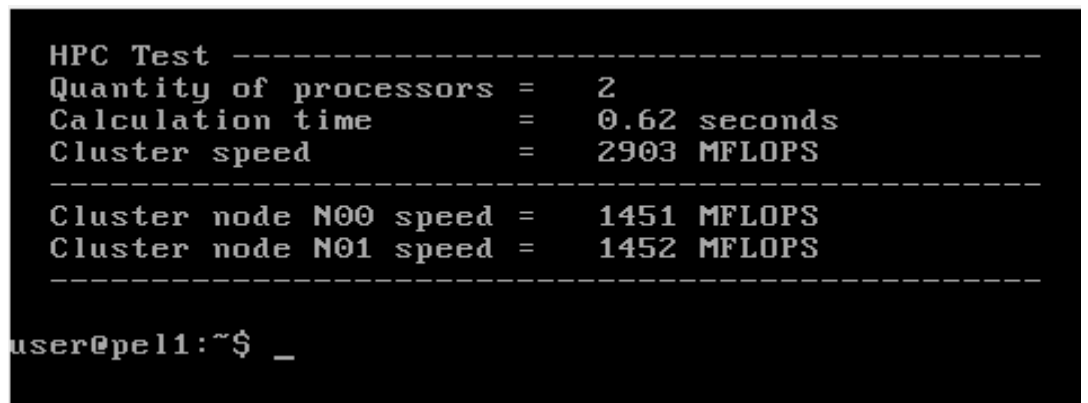
1. Máquina virtual

1.1. Un nodo



Tiempo de cálculo- 15.701 sec.

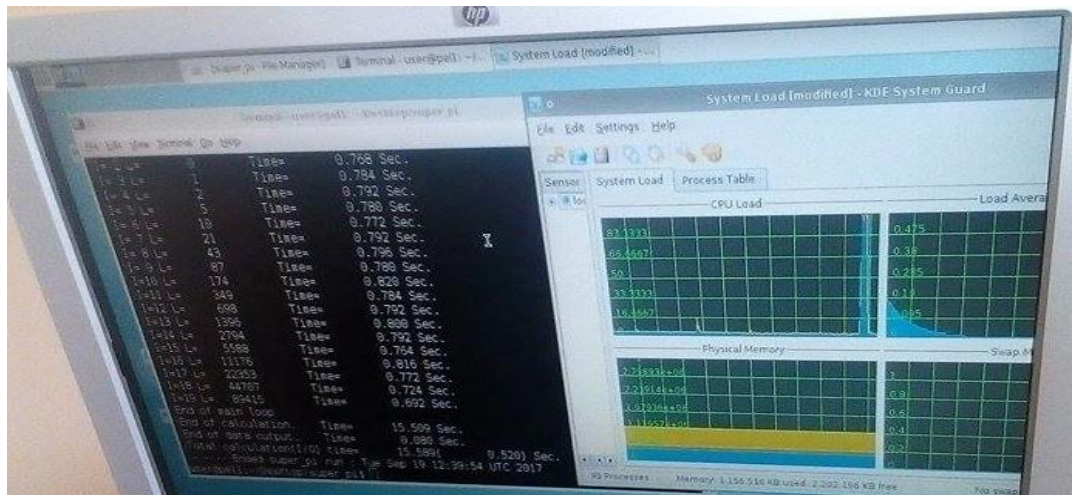
1.2. Dos nodos



Tiempo de cálculo- 13.809 sec.

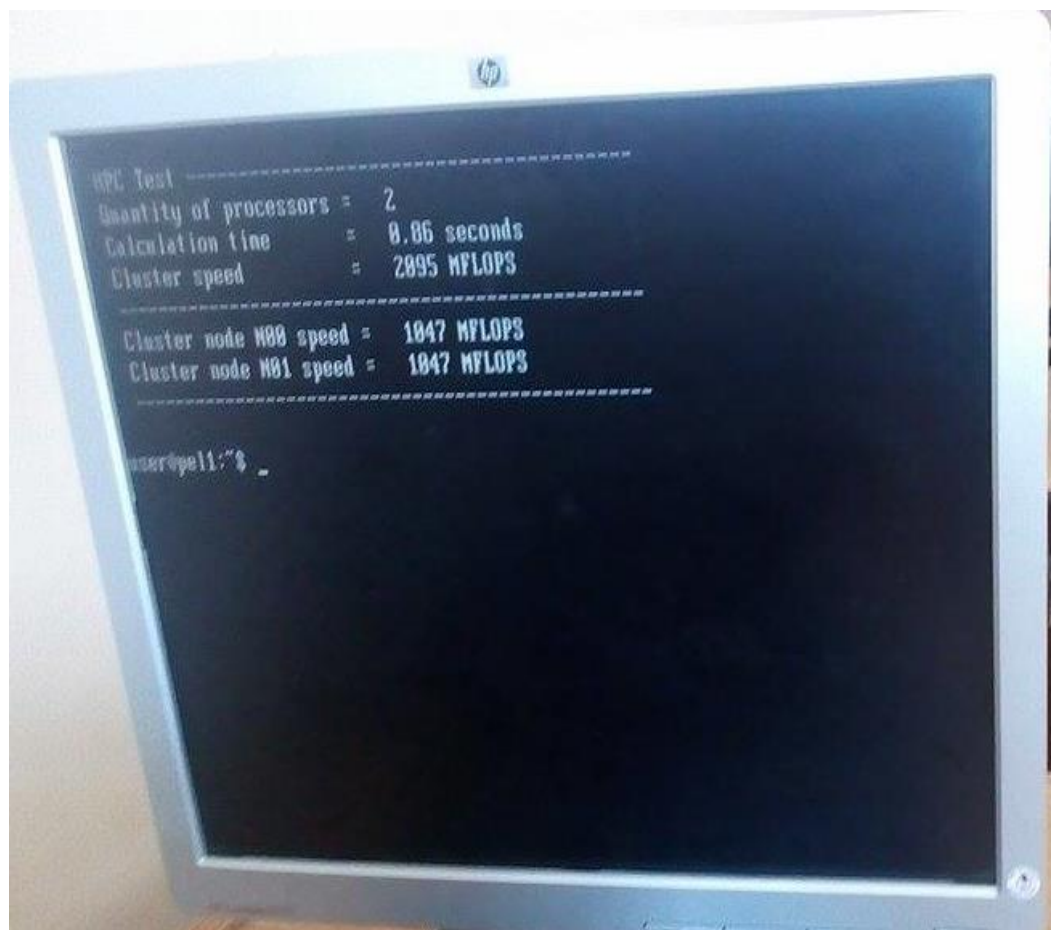
2. Máquina virtual

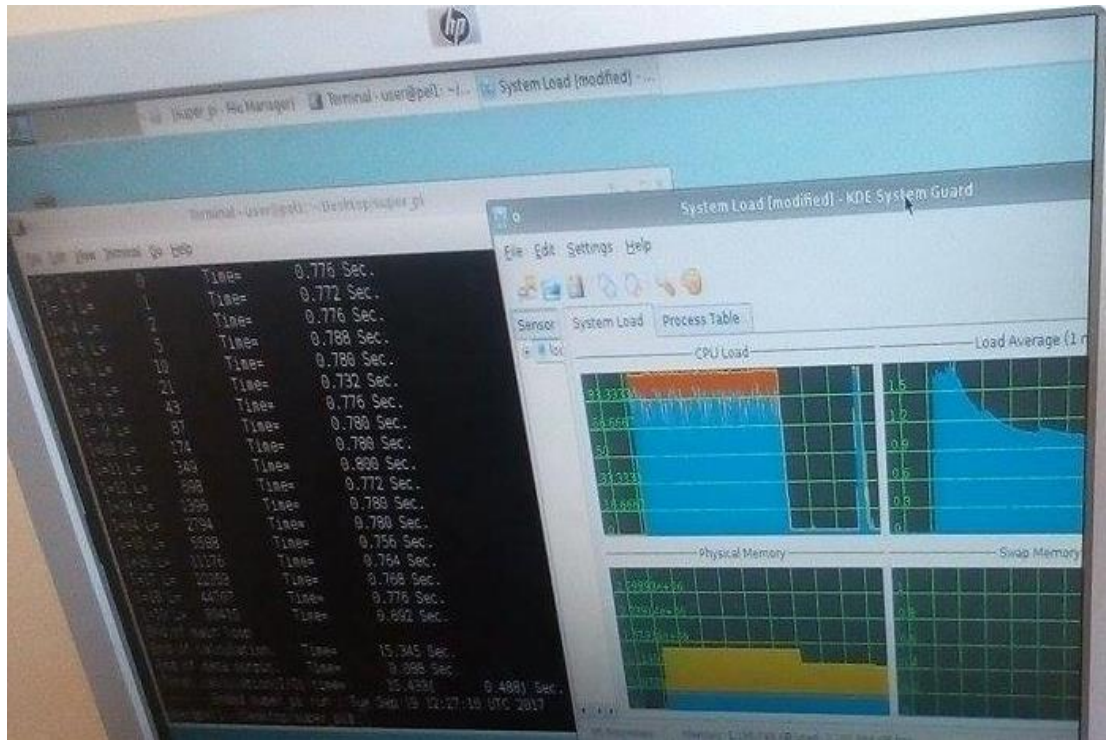
2.1. Un nodo



3. Tiempo de cálculo- 15.509 sec.

3.1. Dos nodos





Tiempo de cálculo- 15.345 sec.

Nodo cliente accediendo al servidor con cuenta de super-usuario



Usar la copia de configuración del usuario poner YES

```

Pelican Setup
Copy user configuration and examples to
/home/user? You should probably choose YES
unless you are using a specially made version
of PelicanHPC that uses a permanent storage
device, and you already have a setup you would
like to keep.
< Yes > < No >

```

En esta parte hay debería reconocer los nodos

```

Restart Pelican HPC
Go turn on your compute nodes now.
At the moment 1 compute nodes (not counting
this frontend node) are available.
Click no to rescan the available nodes. Click
yes when the desired number of nodes are
available. You might want to wait a bit if
some nodes are still finishing booting up.
< Yes > < No >

```

Y en esta otra parte ya debería reconocer todas las pc conectas al switch con todo el servidor

```

HPC Test -----
Quantity of processors = 2
Calculation time = 0.88 seconds
Cluster speed = 2039 MFLOPS
-----
Cluster node N00 speed = 1019 MFLOPS
Cluster node N01 speed = 1019 MFLOPS
-----
user@pel1:~$ _

```

Si no resulta así se vuelve a realizarlo si es que no lo reconoce.

NOTA: es recomendable después de prender el servidor que se vaya prendiendo los nodos, pero configurándolo **MODO DE ARRANQUE POR RED.**