Pontificia Universidad Católica del Ecuador

FACULTAD DE INGENIERÍA
COORDINACIÓN DE POSGRADO

PUCE

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

Trabajo de Titulación como requisito previo para la obtención del título de
Magíster en Tecnologías de Información mención Gestión y Administración de TI

Análisis de los requisitos para una plataforma HPC (High Performance Computing), características, beneficios y campos de aplicación.

Autor: Álvaro Paúl Carrera Carriel

Director: Santiago Silva Proaño

Quito, 2023.

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

DECLARACIÓN Y AUTORIZACIÓN

YO, ALVARO PAUL CARRERA CARRIEL, con C.I. 091961912-2, autor del trabajo de titulación "Análisis de los requisitos para una plataforma HPC (High Performance Computing), características, beneficios y campos de aplicación.". En mi carácter de estudiante de maestría, declaro y autorizo revisar el presente trabajo de titulación.

También, autorizo a la Pontificia Universidad Católica del Ecuador a conservar una copia de la tesis en su repositorio de la biblioteca de la PUCE, con la finalidad de proporcionar el conocimiento adquirido hacia los demás estudiantes, profesores y personal de interés.

ALVARO PAUL CARRERA CARRIEL

0919619122

APROBACIÓN DEL TUTOR

En mi carácter de Director (a) – Tutor (a) del Trabajo de Posgrado Titulado: "**Análisis de los requisitos para una plataforma HPC (High Performance Computing), características, beneficios y campos de aplicación**", presentado por el maestrante ÁLVARO PAÚL CARRERA CARRIEL, titular de la Cédula de Identidad N.º 0919619122 para optar al Grado de Magíster en Educación mención gestión del aprendizaje mediado por TIC, considero que dicho Trabajo de Investigación reúne los requisitos y méritos suficientes para ser sometido a la evaluación por parte de los Lectores – Evaluadores que se designen para tal fin por parte de las autoridades de la Facultad de Ciencias de la Educación.

En la ciudad de Quito, a los … días de ….de 202...

_____

Santiago David Silva Proaño C.I. 171481593-1

ssilva068@puce.edu.ec

NRO TELEFONO: 0992388692

NOTA:

Se comunica que en el servicio de análisis Turnitin, el referido trabajo de titulación alcanzó el siguiente resultado: … % índice de similitud con otras fuentes.

TURNITIN: INCLUIR HOJA DEL INFORME CON EL PORCENTAJE

Tesis HPC

ORIGINALITY REPORT

6%	6%	1%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.coursehero.com Internet Source	1%
2	community.rstudio.com Internet Source	<1%
3	groups.io Internet Source	<1%
4	gitlab.rc.uab.edu Internet Source	<1%
5	repositorio.puce.edu.ec Internet Source	<1%
6	Submitted to Pontificia Universidad Catolica del Ecuador - PUCE Student Paper	<1%
7	Submitted to Universidad Carlos III de Madrid Student Paper	<1%
8	www.renata.edu.co Internet Source	<1%
9	sizin.tistory.com Internet Source	<1%

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, Álvaro Paul Carrera Carriel, como correspondiente autor del trabajo de titulación titulado "Análisis de los requisitos para una plataforma HPC (High Performance Computing), características, beneficios y campos de aplicación" presentado para optar por el grado de Magister en Tecnologías de la Información mención Gestión y Administración de TI, procedo con la siguiente declaración de autenticidad y responsabilidad:

1. Declaro que el presente trabajo de titulación goza como resultado de investigación propia y esfuerzo dedicado. Los conceptos, trabajo práctico, análisis, resultados y conclusiones planteados en el presente documento son de mi autoría, salvo cuando se indique lo contrario en una referencia conveniente con su correspondiente fuente de cita.
2. También, declaro que este trabajo de titulación es original dado que ha sido desarrollado con ética y respeto de los principios de integridad académica. Las referencias, citas, documentos o herramientas informáticas utilizadas de otras fuentes poseen su debido reconocimiento siguiendo las normas establecidas por la institución.
3. Por consiguiente, he procedido con ética bajo los principios de integridad académica para el desarrollo del trabajo de titulación, conservando siempre una honestidad intelectual con estricto parámetro científico a lo largo del transcurso proceso investigativo.

ALVARO PAUL CARRERA CARRIEL

DEDICATORIA

Dedico este trabajo de titulación a Dios, por otorgarme el don de la vida.

A mis profesores de maestría y tutor de tesis por su paciencia y dedicación altruista en impartir su conocimiento académico a lo largo de este proceso de trabajo de titulación.

A mis padres Rosa Carriel y Rafael Carrera por brindarme su apoyo incondicional y aconsejarme en mis momentos duros de todo mi tiempo de vida.

A mis hermanos Karina, Rafael y Zulina por su ejemplo de lucha y perseverancia ante las adversidades presentadas en un país extranjero.

Álvaro Paul Carrera Carriel

ÍNDICE DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE GRÁFICOS

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE INGENIERÍA
MAESTRIA EN TECNOLIGÍAS DE LA INFORMACIÓN MENCIÓN GESTIÓN
Y ADMINISTRACIÓN DE TI

Análisis de los requisitos para una plataforma HPC (High Performance Computing), características, beneficios y campos de aplicación.

Autor: Álvaro Paul Carrera Carriel

Director -Tutor: Santiago Silva P.

Fecha: 30 de agosto de 2023

RESUMEN

En la medida que los datos se incrementan a nivel exponencial, la necesidad de resolver problemas complejos requiere de cómputo masivo. Considerando esta situación, la HPC (por sus siglas en ingles High Performance Computing) facilita el procesamiento de grandes volúmenes de datos, de una forma más veloz que una computadora convencional. Las soluciones HPC ejecutan grandes modelos de cálculos analíticos en infraestructuras que alojan supercomputadoras para emplearlos en el campo médico, científico, automotriz, gubernamental, etc.

Un grid está compuesto por un grupo de ordenadores conectados entre sí en una red informática. El clúster HPC agrupa varios servidores informáticos que son nodos de alta velocidad interconectadas entre sí y que a su vez gestiona las cargas de trabajo. Un nodo se encarga de ejecutar uno o varios servidores de aplicaciones.

Este trabajo de titulación aspira a realizar un análisis de desempeño de una plataforma HPC y comparar sus características, beneficios y campos de aplicación, con la finalidad de otorgar conceptos básicos y fundamentales sobre el uso de herramientas para configurar un clúster HPC. También se realiza un breve análisis de las métricas de desempeño de la plataforma HPC y de la simulación de aproximación del número pi con el objetivo de proveer un conocimiento alentador sobre el uso de centros HPC para investigadores, estudiantes y público en general que fomenten la investigación científica.

Palabras Clave: HPC, Grid, Clúster, Supercomputadoras.

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE INGENIERÍA
MAESTRIA EN TECNOLIGÍAS DE LA INFORMACIÓN MENCIÓN GESTIÓN
Y ADMINISTRACIÓN DE TI

Analysis of the requirements for an HPC (High Performance Computing) platform, characteristics, benefits and fields of application.

Author: Álvaro Paul Carrera Carriel

Director -Tutor: Santiago Silva P.

Date: August 30, 2023

ABSTRACT

As data increases exponentially, the need to solve complex problems requires massive computing. Considering this situation, HPC (High Performance Computing) facilitates the processing of large volumes of data faster than a conventional computer. HPC solutions run large analytical calculation models on infrastructures that host supercomputers for use in the medical fields, scientific, automotive, government, etc.

A grid is made up of a group of computers connected to each other in a computer network. The HPC cluster groups several computer servers that are high-speed nodes interconnected with each other and which in turn manages the workloads. A node is responsible for running one or more application servers.

This present degree work aims to carry out a performance analysis of an HPC platform and compare its characteristics, benefits and fields of application, in order to provide basic and fundamental concepts on the use of tools to configure an HPC cluster. A brief analysis of the performance metrics of the HPC platform and the approximation simulation of the pi number is also carried out with the aim of providing encouraging knowledge about the use of HPC centers for researchers, students and the general public that promote research. scientific.

Keywords: HPC, Grid, Cluster, Supercomputers.

INTRODUCCIÓN

La High Performance Computing (HPC) por sus siglas en inglés o también conocida como computación de alto rendimiento se encarga de ejecutar procesos aritméticos complejos que son capaces de llevarse a cabo solo en supercomputadoras. Aquellos procesos aritméticos requieren del uso de supercomputadoras para dar respuesta a soluciones complejas en distintos escenarios. Estos pueden ser informática militar, entornos climáticos y sismológicos, estudios sobre el genoma humano e infraestructura terrestre de geolocalización, etc.

Con el avance de algoritmos aritméticos que simplifican su ejecución. La HPC permite que la simulación de aquellos algoritmos se lo haga de manera autónoma para generar resultados óptimos en su escenario de aplicación que se lo realice.

Con nuevas tecnologías innovadoras que implican investigación continua, diseño, desarrollo, componentes y sistemas tecnológicamente superiores. Los modelos para las simulaciones basadas en la ciencia permiten generar soluciones que antes se podría decir que eran muy difíciles de llevarlas a cabo. Esto permite a través de un software paralelo avanzado ejecutar algoritmos de manera muy rápida, lo que genera una ventaja competitiva para el escenario actual en el que vivimos.

La informática en alto rendimiento (HPC) es un conjunto de buenas prácticas productivas sobre algoritmos, arquitecturas y tecnologías para realizar tareas de informática científica sobre flujos de trabajos más exigentes con el avance tecnológico actual.

Los entornos en áreas de aplicación de la HPC incluyen modelos de simulación de datos en el diseño de armas nucleares, ámbito científico, métodos de predicción de fenómenos naturales e inteligencia militar para apoyo a la seguridad nacional. El rendimiento de la HPC en dichas áreas de aplicación ha generado un tiempo de solución eficiente para el avance tecnológico.

La HPC es de vital ayuda en el enfoque de la investigación informática mediante los entornos anteriormente mencionados para aprovechar los paradigmas informáticos novedosos. (James A. Ang, 2022).

Para el desarrollo del presente trabajo de titulación se procederá a realizar un análisis de desempeño de un clúster HPC que muestra el uso de recursos y carga de la red mediante la aplicación de Ganglia Monitoring. También, la ejecución del script de algoritmo matemático de aproximación del número *Pi* para verificar su simulación a través del entorno de RStudio Server.

CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

1.1. Formulación del problema

En la actualidad las computadoras tradicionales ejecutan cargas de procesamiento convencional acorde a las necesidades de las personas. La constante aparición de problemas complejos científicos que las computadoras convencionales no son capaces de resolverlas. La HPC [1]saca provecho en procesar grandes volúmenes de datos para mejorar la calidad de vida del ser humano. La evolución tecnológica en volúmenes de datos comerciales o infraestructuras de ciudades inteligentes es necesario dominar la computación en alto rendimiento.

A nivel mundial se puede conseguir computadoras convencionales que el usuario lo encuentra en distintas gamas acorde a las necesidades requeridas. La diferencia entre una computadora convencional y una computadora de alto rendimiento HPC es en la ejecución de procedimientos de operaciones de cálculos complejos como lo son del tipo científicos, analíticos o de ingeniería. Debido a los altos costos de inversión que acarrea implantar los entornos HPC y el manejo de gran cantidad de datos que hay que procesar se recurre a la computación HPC para que los pueda realizar.

De acuerdo con (James A. Ang, 2022) “La informática de alto rendimiento ha sido, y continúa siendo, una tecnología habilitadora clave para el diseño y la prueba de I+D de sistemas nuevos y mejorados. El Plan de estrategia militar nacional de Estados Unidos depende de la capacidad de la comunidad científica y de ingeniería de dicha nación para continuar investigando, diseñando, desarrollando y produciendo ideas, componentes y sistemas tecnológicamente superiores.”

Entre las situaciones que se identificaron como puntos críticos en los proyectos de High

[1] HPC High Performance Computing

Performance Computing tenemos:

a) La falta de ejecución de grandes volúmenes de datos en la computación paralela o distribuida.
b) Su implementación es muy costosa y requiere de mucho conocimiento.
c) La necesidad de adaptación de los sistemas informáticos en un entorno de clúster HPC para los cálculos aritméticos adecuados.
d) Los data center requieren de una infraestructura de climatización óptima para la ejecución de los altos volúmenes de datos.

En la región de Latinoamérica, los centros de procesamiento de datos tipo HPC son muy limitados debido a su alto costo de implementación y uso del manejo de datos para el desarrollo investigativo que es un factor considerable por los centros de investigación que apuestan sobre la tecnología en HPC. Referente a Centroamérica, su principal centro de estudios para la HPC se encuentra en México en la ciudad de Guadalajara. En Sudamérica se encuentra en Brasil en la ciudad de Sao Paulo. Con respecto a los centros HPC en el Ecuador, los únicos centros en poseer entornos HPC son la Universidad San Francisco de Quito, la ESPE y el CEDIA (Corporación Ecuatoriana para el Desarrollo de la investigación y la Academia).

1.2. Objetivos de la Investigación

1.2.1. Objetivo General

Analizar los requisitos para una plataforma HPC (High Performance Computing), características, beneficios y campos de aplicación.

1.2.2. Objetivos Específicos

- Recabar información sobre los requerimientos de una plataforma HPC que permita identificar sus características, beneficios y campos de aplicación.

- Construir un clúster virtualizado mediante la herramienta de configuración OpenHPC.
- Realizar el análisis de desempeño de un clúster mediante la aplicación Ganglia Monitoring.
- Efectuar la ejecución de aproximación del número pi en el entorno RStudio Server para verificar su simulación.

1.3. Justificación de la Investigación

El siguiente trabajo de titulación está dirigido al análisis de los requisitos de una plataforma HPC, sus características, beneficios y campos de aplicación. La razón para realizar el presente trabajo de titulación es debido a la escasa implementación de centros HPC que fomenten el desarrollo investigativo en las instituciones para resolver problemas complejos relacionados con el procesamiento acelerado en la ejecución de tareas. Con el avance de las infraestructuras de TI en aplicaciones, simulaciones, mediciones y cálculos; se requieren opciones de cómputo en paralelo para lograr procesos óptimos que generen resultados favorables hacia el avance científico y tecnológico de la humanidad en general.

Referente a lo anterior, la tecnología HPC busca ejecutar procesos de manera paralela que brinden soluciones a problemas complejos en el ámbito científico, industrial, medico, físico, químico y meteorológico. Motivo por el cual se realiza este presente trabajo de titulación.

CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA

2.1. Antecedentes de la Investigación

Para (Rivera, 2017) en su proyecto de titulación cuyo objetivo principal fue el de analizar la escalabilidad del modelo de predicción climática WRF en clústeres HPC en base a las comunicaciones y procesos MPI[2]. Entre sus resultados obtuvo un desempeño máximo en su red Infiniband de 25.9, QDR de 25.9 y Ethernet de 6.4 con referencia a una ejecución secuencial como parte de las configuraciones para aumentar su desempeño. Consiguió una conclusión de ser una opción favorable para los requerimientos de su plataforma porque están enfocados hacia la portabilidad.

(Andrés, 2021) en su proyecto de tesis tuvo como objetivo realizar un estudio sobre la paralelización de ACO mediante MPI (por sus siglas en ingles Message Passing Interface) y programación de GPU haciendo uso de la plataforma CUDA con un diseño de análisis de la paralelización de dos componentes principales del algoritmo con una muestra de 48 procesadores. Para sus resultados se basó en las métricas de speedup, eficiencia y Karp-Flatt para evaluar el MPI con la tecnología de HPC.

[2] ¿Qué es MPI? (Message Passing Interface) Interfaz de mapeo de mensajes.

2.2. High Performance Computing (HPC)

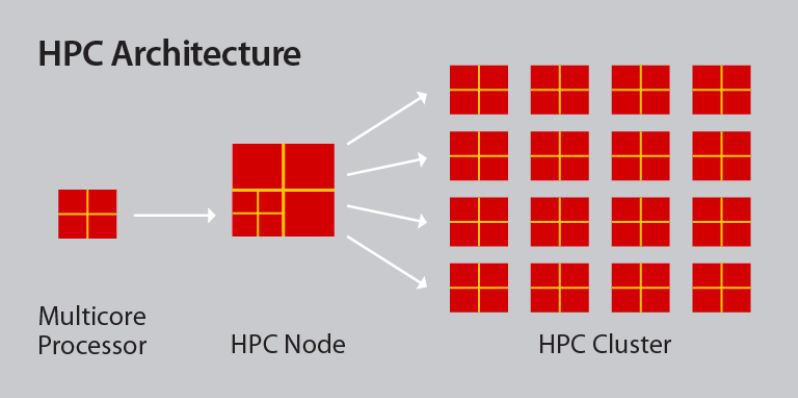
2.2.1 Que es High Performance Computing

Como lo define (Corral García, 2021) en su tesis doctoral, la HPC es una supercomputación que requiere de un rendimiento amplio sobre las computadoras convencionales que no son capaces de realizar resoluciones de problemas científicos, en el campo de ingeniería o de empresa. Con esta tecnología de supercomputación se puede obtener resultados satisfactorios sobre una variedad de escenarios complejos que no se los puede resolver de manera empírica o teórica en ordenadores simples. Factores de requerimientos como almacenamiento, memoria o computación hace que la HPC en la era moderna presente impactos favorables en los diferentes campos de aplicación para el desarrollo progresivo de la humanidad actualmente.

La función de una arquitectura HPC radica en que los servidores informáticos se reconectan a una red de clúster que se explica a detalle en el apartado 2.3. lo que es un clúster. Las aplicaciones y algoritmos se ejecutan de forma simultánea en los servidores de clústeres que están conectados al almacenamiento de datos para realizar las distintas tareas que se vayan a emplear. A continuación, mediante el siguiente gráfico se muestra cómo funciona una arquitectura HPC. (RCAB, 2023)

Se observa a continuación en el Gráfico 1 el funcionamiento de una arquitectura de entorno HPC, donde el nodo de cómputo del servidor posee un procesador multinúcleo interconectado entre sí para conformar lo que es el clúster HPC.

Gráfico 1
Funcionamiento de Arquitectura HPC



Fuente: (RCAB, 2023)

2.2.2. *Características HPC*

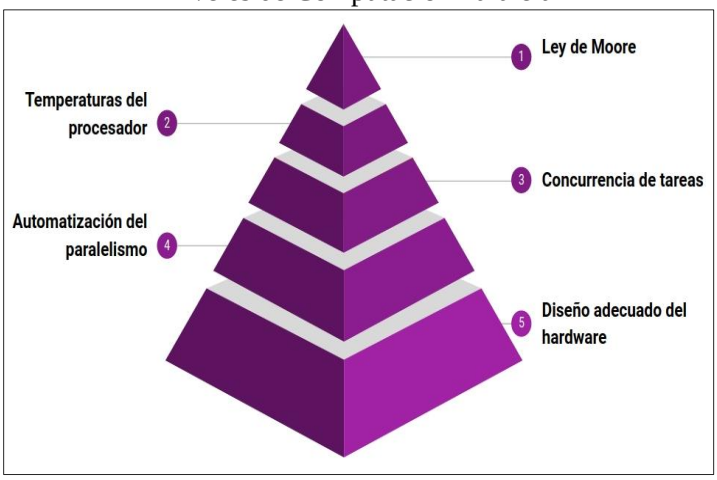
Entre las principales características de HPC se pueden destacar las siguientes:

Tecnología computacional paralela: Se refiere a la ejecución de varias tareas de manera simultánea en servidores informáticos masivos que permite el uso de miles o hasta millones de procesadores.

Como se puede apreciar en el siguiente Gráfico 2 cada uno de los niveles de recursos que utiliza la computación paralela, en el nivel 1 hace uso de la ley de Moore que potencia los componentes informáticos haciéndolos más reducidos y de rápida ejecución. En el nivel 2 realiza una refrigeración permanente y óptima del procesador. Con el nivel 3 se procede con la concurrencia de tareas que trabajan al mismo tiempo. Para el nivel 4 se adopta estrategias para la automatización de procesos de una manera proporcional al número de núcleos; y en el nivel 5 se adopta una solución en el hardware que permite la paralelización acorde a los procesadores que se estén utilizando. (Fabián Bernal, 2020)

Gráfico 2

Niveles de Computación Paralela



Fuente: (Fabián Bernal, 2021)

Conjunto de Clústeres: Es un esquema en el que coexisten varios clústeres HPC dentro de servidores informáticos conectados entre sí por medio de una red de comunicación. Se encarga de gestionar la carga de trabajo de cómputo en paralelo, que, a su vez, está compuesta de nodos de ordenadores para la ejecución de tareas. A continuación, se divisa en el siguiente Gráfico 3 el centro HPC del laboratorio de innovación de la empresa Dell Technologies.

Gráfico 3

Laboratorio de innovación de HPC e IA de Dell Technologies

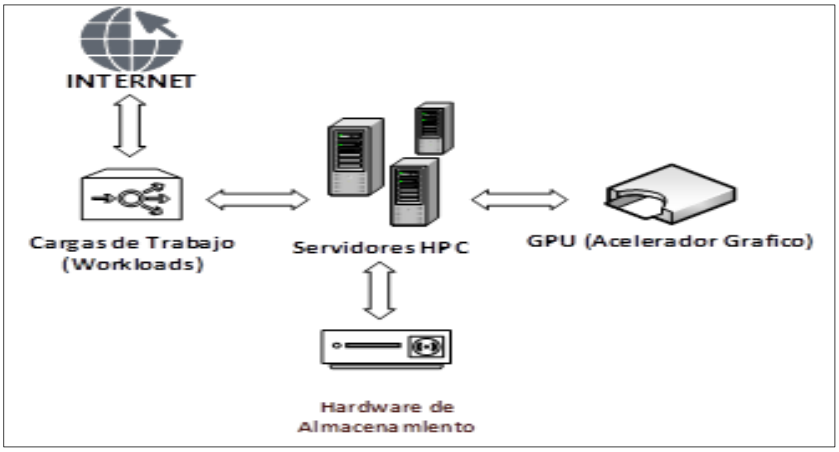
Fuente: (DataCenterMarket, 2022)

Alto rendimiento en tecnología computacional: Implica recursos informáticos de un clúster HPC como almacenamiento, redes, memoria y sistemas de archivos. Estos exigen una alta velocidad en ejecución y baja latencia en sus nodos para la optimización y rendimiento del clúster. (IBM, 2023)

Se muestra en el siguiente Gráfico 4 cada uno de los recursos informáticos que conforman un servidor HPC.

Gráfico 4

Uso de Recursos de un Sistema HPC



Fuente: El Autor

2.2.3. Beneficios

Dentro de los beneficios de la tecnología HPC se destaca los indicados a continuación en la Tabla 1.

Tabla 1

Beneficios de HPC

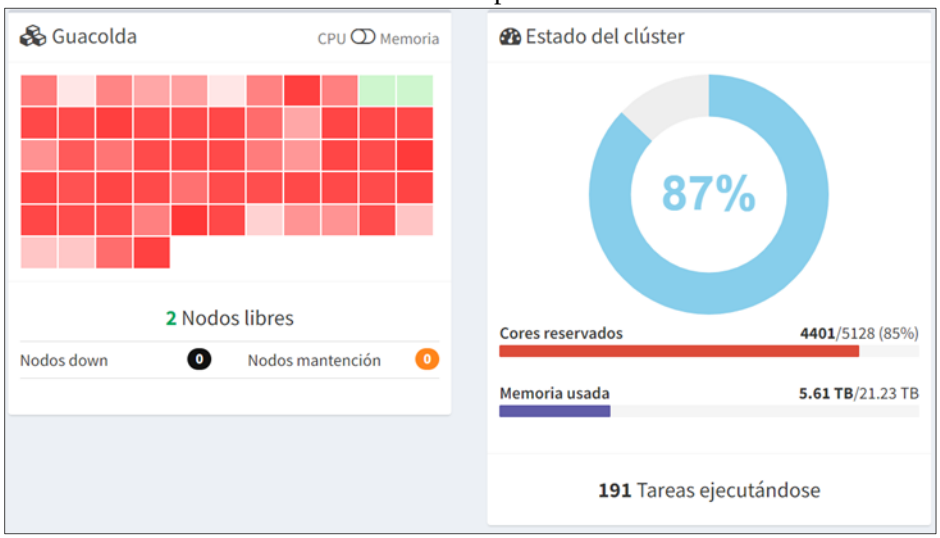
Beneficios High Performance Computing	
Beneficio	**Detalle**
Reducción en pruebas de	Se puede utilizar para simulaciones de diversos ámbitos que facilita la reducción de pruebas consecutivas para determinar un resultado obtenido.

cómputo	
Novedoso	Impulsa la innovación hacia la investigación en ciertos sectores del campo científico. Esto revoluciona contantemente los desafíos de la humanidad mejorando su calidad de vida en la actualidad.
Prontitud	Con las nuevas tecnologías en GPU (Unidades de procesamiento gráfico) se obtiene una carga de red de menos latencia que optimiza los accesos a memoria para alcanzar respuestas veloces a los cálculos que se realicen.
Reducción de Costos	A mayor demanda en tiempos de respuestas se puede evitar el desperdicio de tiempo y dinero para la ejecución de cargas de trabajo de HPC que brinda una libre elección del servicio según la necesidad que se requiera.

Fuente: (Oracle , 2023)

Se divisa en el siguiente Gráfico 5 el beneficio del funcionamiento de un clúster HPC con cada uno de sus nodos, el estado de sus cores y su memoria.

Gráfico 5
Beneficio de Desempeño del Cluster



Fuente: (NLHPC, 2023)

2.2.4. Campos de Aplicación

Para la presente Tabla 2 se describe cada uno de los campos donde se aplica la HPC acorde al entorno en el que se lo requiera implementar.

Tabla 2

Campos de Aplicación de la HPC

Campos de Aplicación HPC	
Campo	**Descripción**
Empresarial	Se aplica modelos financieros por medio de la ciencia de datos para emplear tecnologías de inteligencia artificial y aprendizaje automático porque es un motor clave para las empresas en capital e innovación.
Gubernamental	Se realiza estudios sobre el impacto ambiental para la investigación científica en el uso de materia energética, defensa nuclear o ataque biológico, etc.
Médico	Se aplica para crear vacunas, investigar sobre nuevos medicamentos y desarrollar tratamientos medicinales para la innovación médica.
Investigativo	Se lo emplea para la implementación de chips en la industria farmacéutica en materia de laboratorios de dinámica molecular.
Manufactura	Se aplica en la simulación de la ingeniería para la rama de la industria que utiliza la robótica con diseños avanzados en la mejora de su automatización.
Colaborativo	Genera soluciones que aplican la computación cuántica para desarrollar modelos de programación paralela, fomentando la investigación o descubrimientos biomédicos.
Centro de datos	Se emplea soluciones HPC en la nube para favorecer el acceso a la información que se optimiza por medio de cargas de trabajo en el ámbito de negocios o defensa gubernamental.

Fuente: (InsideHPC, 2023)

Se muestra a continuación en el Gráfico 6 un ejemplo de las empresas miembros de la Comunidad OpenHPC que hacen uso de la plataforma a nivel mundial.

Gráfico 6

Empresas Miembros de la Comunidad OpenHPC

Fuente: (openHPC, 2020)

2.2.5. HPC en Latinoamérica

En el caso de Latinoamérica se tiene el caso de “RedCLARA” que es una organización que integra una red de representantes de sistemas nacionales de HPC en la región, centros de supercomputación y observación de representantes internacionales que fomenta el fortalecimiento de competencias en desarrollo de proyectos que involucren la computación científica y de alto rendimiento. Lo conforman centros de investigación científica con soluciones HPC para el uso de comunidad de estudiantes, investigadores y tecnólogos, entre ellos, se encuentra el Ecuador por medio de CEDIA (Corporación Ecuatoriana para el Desarrollo de la Investigación y la Academia). (RedCLARA, 2023)

A través de la RedCLARA se conforma también CARLA (Conferencia Latinoamericana de High Performance Computing) conforme a (CARLA, 2023) “se encarga de proporcionar un foro para

fomentar el crecimiento y la fortaleza de la comunidad de Computación de Alto Rendimiento (HPC) en América Latina a través del intercambio y la difusión de nuevas ideas, técnicas e investigaciones en HPC y sus áreas de aplicación.”

2.2.6. Top 500

El top 500 es una página que enlista las 500 supercomputadoras a nivel mundial que mide su rendimiento en base a experiencias de usuarios y proveedores. Su top de lista se actualiza semestralmente para dar seguimiento a la evolución de los ordenadores HPC. Tiene como objetivo proporcionar una lista de sistemas de propósito general para aplicaciones de gama alta que sean capaces de resolver problemas científicos. La medida de desempeño que emplea para el ranking es el HPL (High Performance Linpack Benchmark) que es una aplicación que resuelve un sistema lineal aleatorio de aritmética de doble precisión (64 bits) en computadoras de memoria distribuida. (TOP 500, 2022)

Mediante la Tabla 3 se muestra el top 5 de las supercomputadoras con sus principales características. Se toma en cuenta el ranking con fecha de la última publicación semestral del mes de noviembre del año 2022.

Tabla 3

Top 5 Mejores Supercomputadoras

Ranking 5 Mejores Supercomputadores										
Posición	**Sistema**	**Fabricante**	**País**	**Procesador**	**Cores**	**Rmax (PFlop/s)**	**Rpeak (PFlop/s)**	**Potencia (kW)**	**SO**	**Campo de Aplicación**
1	Frontier	HPE (Hewlett Packard Enterprise)	EEUU	AMD Optimized 3rd Gen 64C 2GHz	8,730,112	1,102.00	1,685.65	21,100.00	HPE Cray OS	Oak Ridge National Laboratory
2	Supercomputer Fugaku	Fujitsu	Japón	A64FX 48C 2.2GHz	7,630,848	442.01	537.21	29,899.23	Red Hat Enterprise Linux	RIKEN Center for Computational Science
3	LUMI	HPE	Finlandia	AMD Optimized 3rd Gen 64C 2GHz	2,220,288	309.10	428.70	6,016	HPE Cray OS	EuroHPC/CSC
4	Leonardo	ATOS	Italia	Xeon Platinum 8358 32C 2.6GHz	1,463,616	174.70	255.75	5,610	Linux	EuroHPC/CINECA
5	Summit	IBM	EEUU	IBM POWER9 22C 3.07GHz	2,414,592	148.60	200.79	10,096	RHEL 7.4	Ridge National Laboratory

Fuente: (Top500, 2022)

*PFlop/s: Operaciones de Punto Flotante por segundo

*Rmax: Rendimiento Máximo Alcanzado

*Rpeak: Rendimiento Máximo Teórico

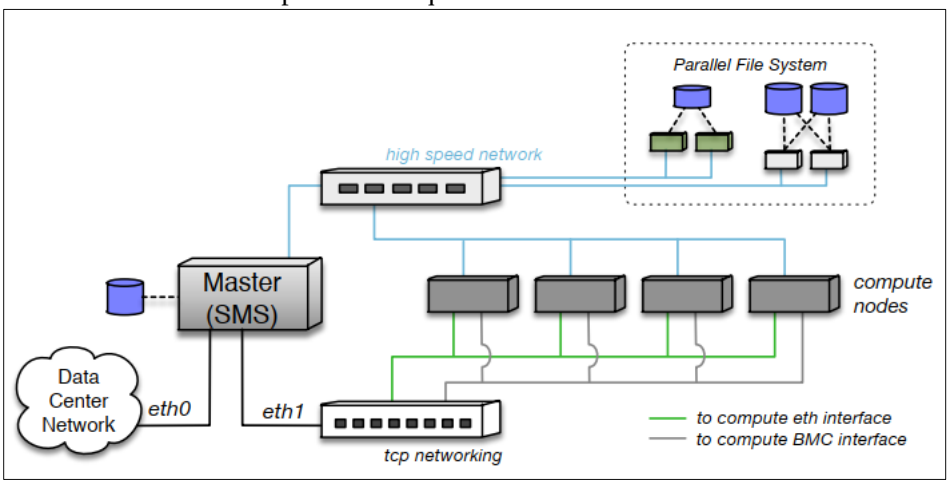
2.3. Clúster HPC

Un clúster es una agrupación de nodos de cómputo autónomos e independientes que se integran en una red privada de alta velocidad, donde cada uno posee su propio sistema operativo. Sus nodos son de funcionamiento básico que alcanzan un rendimiento limitado con respecto a procesadores paralelos masivos. Aunque en conjunto muestran una mejor relación costo-beneficio y una escalabilidad considerable frente a los demás computadores convencionales. (Corral García, 2021, p. 11)

Por otro lado, un cluster HPC agrupa varios servidores informáticos conectados entre sí, que trabajan con grandes tareas para un gran desempeño computacional. Se encarga de gestionar la carga de trabajos masivos en computación paralela que integran CPU de alto rendimiento o GPU (Unidad de procesamiento gráfico) con la particularidad que puede contener alrededor de 1000 nodos o más. (IBM, 2023)

Se divisa en el presente Gráfico 7 una breve descripción de los componentes que conforman un clúster físico HPC.

Gráfico 7
Descripción de Arquitectura de Clúster Físico



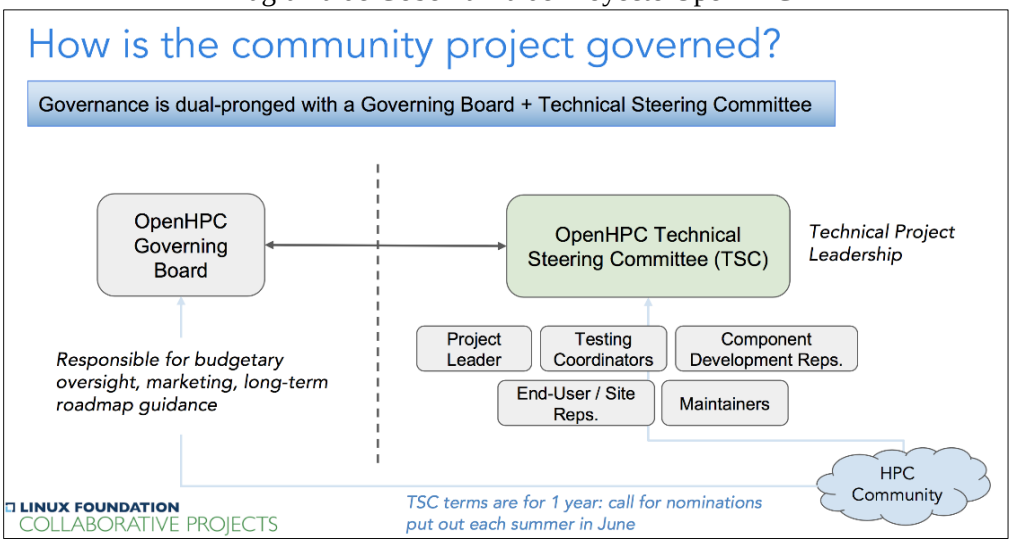
Fuente: (openHPC, 2020)

2.3.1. Open HPC

Open HPC es una herramienta colaborativa *Open Source* para implementar y/o administrar clústeres de distribución Linux para sistemas HPC. Contiene herramientas de provisionamiento, administración de recursos, clientes de E/S, herramientas de desarrollo y una gama de bibliotecas científicas. Además, integra fuentes de fabricantes de equipos, sitios de supercomputadoras e instituciones de investigación. (openHPC, 2020). A continuación, se muestra en el Gráfico 8 como está estructurado el gobierno del proyecto OpenHPC y de cada uno de los actores que lo intervienen.

Gráfico 8

Diagrama de Gobernanza de Proyecto OpenHPC



Fuente: (openHPC, 2020)

2.3.2. Ganglia Monitoring System

Ganglia es un sistema de monitoreo distribuido escalable para sistemas HPC como clústeres y grids[3]. Su diseño es a nivel jerárquico conformado por clústeres que aprovechan las tecnologías utilizadas. Optimiza considerablemente el uso de recursos de los nodos por medio de la estructura

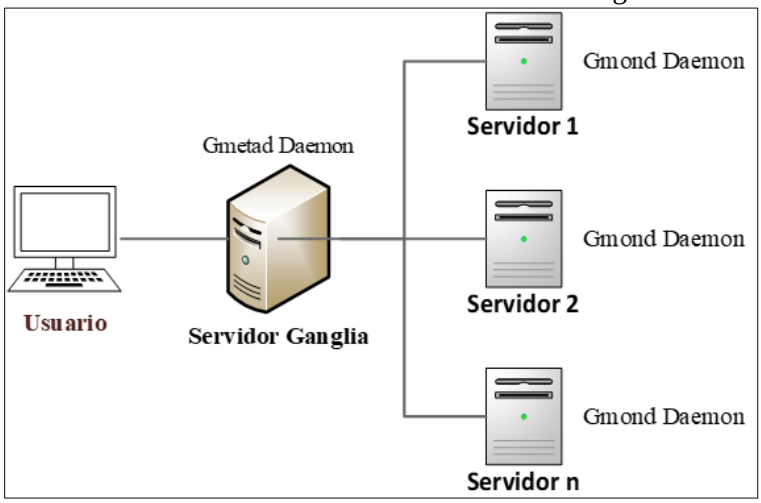
[3] Grid --- Conecta un grupo de ordenadores que forman un superordenador virtual

de datos y algoritmos. Su implementación es robusta porque concentra varios sistemas operativos, arquitecturas de procesador y miles de clústeres a nivel mundial. Es requerido actualmente por laboratorios científicos y campus universitarios. (NVIDIA DEVELOPER, 2023)

A continuación, se divisa en el Gráfico 9 el acceso al sistema por parte de los usuarios a cada uno de los servidores para proceder al monitoreo del clúster desplegado.

Gráfico 9

Acceso al funcionamiento del Sistema Ganglia



Fuente: El Autor[45]

2.4. Virtualización HPC

La virtualización HPC se refiere a la creación de un numero de servidores virtuales que están alojados dentro un servidor físico u host. Añade flexibilidad, eficiencia operativa, agilidad y seguridad para ejecutar múltiples aplicaciones en una misma máquina que administra los recursos de memoria, CPU y red. Reduce el tiempo considerablemente en la recopilación de información y datos.

Entre las principales características que se pueden encontrar en este tipo de Arquitectura están las

[4] Gmond: Datos que contiene en memoria el servidor.
[5] Gmetad: Consulta periódicamente a los datos Gmond

descritas en la siguiente Tabla 4.

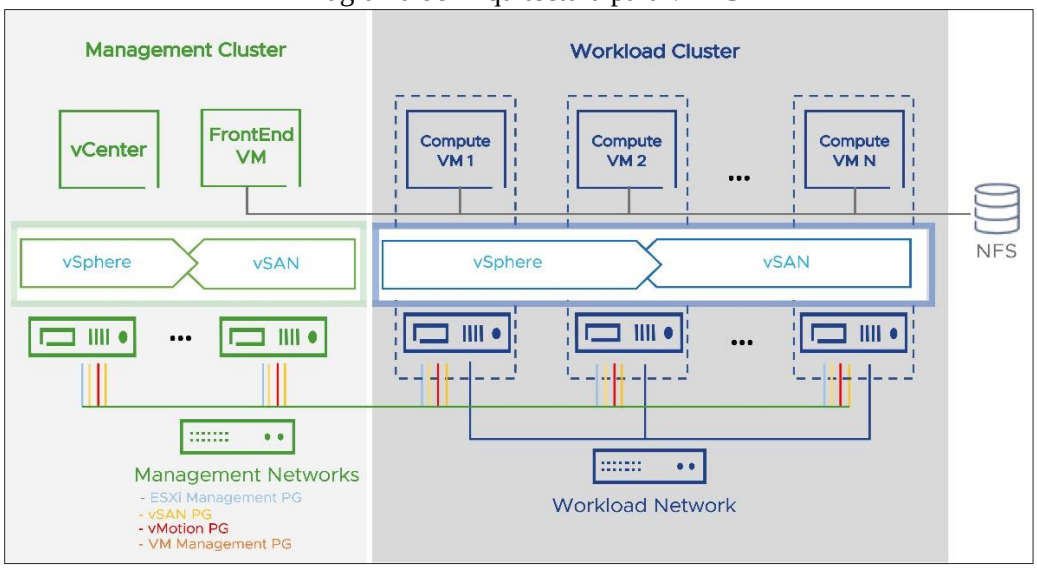
Tabla 4
Características de Virtualización HPC

Características de la Virtualización HPC	
Tipo	**Descripción**
Flexibilidad y Agilidad	Acelera el aprovisionamiento de infraestructuras de TI según las necesidades requeridas para agilizar la iteración. Prioriza más tiempo a obtener información sin mucha demanda a implementaciones o rediseños.
Eficiencia Operativa	Se reduce considerablemente el tiempo de instalación y configuración para prestaciones de gestión centralizada. Optimiza las operaciones de aprovisionamiento y mantenimiento de infraestructuras.
Menor Complejidad	Su aporte significativo en la oferta de servicios de valor en copias y análisis de seguridad, cortafuegos y gestión de operaciones.
Control de datos y confidencialidad de los clientes	Cada máquina virtual se blinda en seguridad porque forma parte del mismo clúster para acceder a los archivos que correspondan.

Fuente: (vmWARE, 2023)

A continuación, se visualiza en el Gráfico 10 el diagrama de arquitectura HPC virtualizada.

Gráfico 10
Diagrama de Arquitectura para vHPC



Fuente: (Yuankun Fu, 2022)

2.5. Paralelización de Cómputo HPC

2.5.1. Concurrencia y Paralelización de algoritmos

Este proceso se realiza con la ejecución de dos o más tareas de cómputo independientes dentro de un mismo lapso. Por lo regular, es utilizado en sistemas multitareas donde un único núcleo aplica una técnica de división para que se ejecuten de manera simultánea para un paralelismo virtual. Con referencia al análisis y diseños de algoritmos se considera la descomposición y asignación de tareas a las unidades de procesamiento. La metodología PCAM (Partitioning, Communication, Agglomeration and Mapping) presenta una paralelización funcional que realiza una descomposición natural de las operaciones con respecto a objetivos computacionales independientes.

La paralelización de códigos secuenciales se centra en optimizar las comunicaciones entre procesos y maximizar la computación. Acorde al modelo de programación que se use, su objetivo es disminuir de manera eficiente los tiempos de ejecución. Las tareas paralelas tienden a variar su

tiempo de ejecución, lo que considera un equilibrio equitativo en la carga para coordinar fragmentos de códigos asociados al problema que se presenta. La paralelización de bucles permite realizar cálculos que recorren iterativamente el algoritmo, es necesario que no exista dependencias entre las instrucciones para las distintas iteraciones y ser ejecutadas sin un orden establecido. (Corral García, 2021, pp. 13,14,16)

2.5.2. Punto Flotante HPC, Número Pi y RStudio

El punto flotante por segundo es la métrica de rendimiento que más se utiliza en la HPC, no obstante, no es la única medida que se refleje en el ámbito de las supercomputadoras. Los sistemas HPC son clasificados en un ranking que basa sus resultados en técnicas de benchmark. Las características del ranking se reflejan con la información del fabricante, número de procesadores y núcleos, memoria, consumo de energía y sistema operativo. El benchmark se enfoca en el rendimiento máximo de ejecución sobre las operaciones de punto flotante para determinados sistemas de ecuaciones que no comprometa la memoria o red interna. (Corral García, 2021, p. 8)

El valor del *número Pi* tiene mucha significancia en los campos del cálculo y la física. Hay variedad de métodos que permiten encontrar su valor aproximado. Para alcanzar su resultado se lo emplea por una serie de sumatorias que se la representa de la siguiente manera.

$$\sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} = \frac{\pi}{4}$$

Dicha expresión concurre en una serie infinita que genera el siguiente resultado de π /4

$$\pi = \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} \ldots$$

La fórmula del *número Pi* necesita de un número n para llegar al valor aproximado del mismo. Con la condición de que, si n es un número cercano a 3,000,000 su valor más próximo será de:

3.14159245358979323846464338327795027841971693991344541111…

Observando este resultado, el mismo se lo puede expresar en un algoritmo o lenguaje de programación para representar el valor del *número Pi*. (Velazquez, 2017)

En ese contexto se puede usar R, que es un lenguaje de programación de ambiente de software orientado al análisis estadístico, representación gráfica y generación de reportes. RStudio es un IDE alojado en un servidor Linux que se accede a través de un navegador web para potenciar la productividad en un entorno centralizado. R para HPC es de mucha utilidad porque ofrece herramientas de paquetes y funciones para realizar procesamiento en paralelo consiguiendo la máxima optimización de recursos. Su experiencia de uso refiere una escalabilidad en resolver problemas, brindar solución y tecnología ajustable. (Alvear, 2018)

CAPÍTULO III: METODOLOGÍA

3.1. Tipo de Investigación

Para el desarrollo del presente proyecto de titulación se utilizó el tipo de investigación cuantitativa que sirvió para el análisis de la plataforma HPC con la elaboración de las pruebas de desempeño.

3.2. Fases del Análisis de la plataforma HPC

Para el análisis de la plataforma HPC de este trabajo de titulación, se determinó una metodología establecida en dos fases, con el objetivo de abarcar la definición de los objetivos para el reconocimiento de los requisitos planteados.

3.2.1. Instalación de componentes y construcción del clúster virtualizado

Para esta primera fase se procede a realizar las respectivas instalaciones de componentes del sistema HPC incluyendo las herramientas de aprovisionamiento. Posteriormente, se realizó la construcción del clúster virtual para analizar el desempeño del mismo.

3.2.2. Ejecución del programa de aproximación del número pi con sus respectivas pruebas

Ejecutado el diseño del clúster en la fase anterior se procedió a realizar la ejecución del programa de aproximación del *número Pi* para determinar las pruebas obtenidas de su simulación. En esta fase se trabaja de manera iterativa para comprobar la aproximación del *número Pi* y verificar el desempeño del clúster con sus respectivas métricas.

3.3. Posibles Limitaciones

Este trabajo de titulación se enfocó en el uso de las herramientas Open HPC, vmWARE y RStudio. Debido a que la plataforma HPC exige bastantes recursos para su funcionamiento, se recurrió a la herramienta de virtualización para optimizar la construcción del clúster, aunque no se refleje mejoras en la paralelización debido a que se está ejecutando en un solo procesador físico. De igual

manera, se hace uso de una arquitectura básica del clúster que consiste en usar un solo Grid, que es un nodo master y dos nodos cómputo. La mayoría de los laboratorios de HPC a nivel mundial tienen acceso restringido a sus plataformas debido básicamente al tema de seguridad en su infraestructura de trabajos en HPC, solo permiten el ingreso previo a solicitud por parte de los encargados del centro de HPC en las instituciones.

Para la realización de las pruebas de aproximación del *número Pi*, el acceso a los entornos de laboratorios HPC tiene la limitante de ser de uso exclusivo de sus usuarios, por lo que se recurrió al uso del entorno RStudio para realizar las pruebas de simulación correspondientes.

3.4. Herramientas de Apoyo para el Análisis

Mediante la aplicación de Ganglia se realizó el análisis de desempeño de la plataforma con una de sus principales métricas que es la de carga del servidor. El código del script de aproximación del *número Pi* se encuentra disponible en el repositorio de GitHub de la Universidad del Sureste de California (USC).

3.5. Métodos de Validación

Para validar las pruebas de desempeño y simulación del *número Pi* se realizó ejecuciones de manera local. Como se lo planteó en la primera fase detallada en el apartado 3.2.1 se procedió a la configuración del nodo master donde se alojó la plataforma OpenHPC. Para garantizar su respectiva prueba de desempeño se ejecutó el script del *número Pi* desarrollado en lenguaje de programación R y ser ejecutado en el entorno RStudio Server.

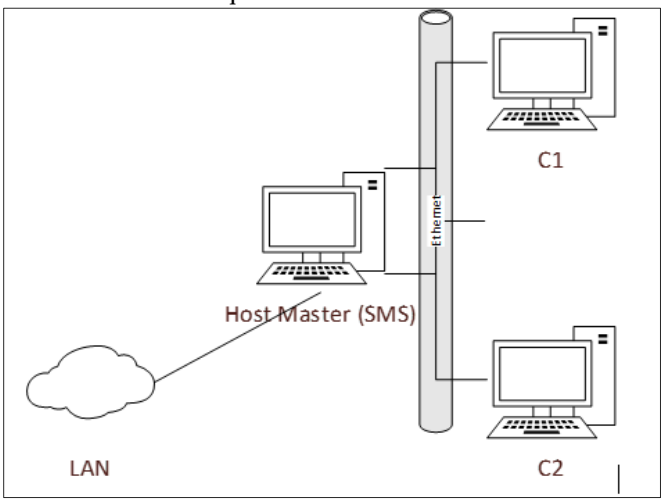
CAPÍTULO IV: PRESENTACIÓN Y ANÁLISIS DE LA PLATAFORMA

En este capítulo se explica cómo se implementó el clúster virtualizado en el hipervisor de VMWare y su configuración con la herramienta OpenHPC. Posteriormente, se realizó el análisis de las principales métricas disponibles en la plataforma HPC con la aplicación de Ganglia Monitoring. Por último, se realizó la ejecución del script del *número Pi* en el entorno RStudio Server para verificar su simulación de aproximación.

4.1. Propuesta de la Arquitectura del Clúster HPC

Se presenta en el siguiente Gráfico 11 la arquitectura del clúster virtual compuesta por los nodos host master, C1 y C2 que servirán para su configuración con la herramienta OpenHPC. Los procedimientos para la configuración de los componentes generales del clúster se los puede visualizar en el correspondiente apartado de ANEXO 2.

Gráfico 11
Arquitectura del Clúster



Fuente: El Autor

4.1.1. Especificación de características del Cluster

Se detalla en la siguiente Tabla 5 las características de cada uno de los nodos que conforman el clúster que son los nodos master, C1 y C2 respectivamente.

Tabla 5
Características del Cluster

Máquina Virtual	Hostname	Disco	Memoria RAM	Velocidad de Transmisión	Sistema Operativo
Master	TesisMaster	20 GB	2 GB	1 Gbits/s	CentOS 7x
Cómputo 1	C1	8 GB	1 GB	1 Gbits/s	CentOS 7x
Cómputo 2	C2	8 GB	1 GB	1 Gbits/s	CentOS 7x

Fuente: El Autor

4.1.2. Clúster Virtualizado

El presente clúster se lo desarrolló con la herramienta VMWare Workstation Pro versión 17 para aprovechar la optimización de recursos de manera considerable en la presentación de la propuesta planteada en el inicio del Capítulo 4. Por ser un clúster virtualizado la ejecución del paralelismo se limita debido a que su arquitectura está implementada en un solo procesador, por consiguiente, se recurre al entorno RStudio Server que se encuentra configurado en el nodo master del grid TesisMaster.

Se muestra a continuación en la Tabla 6 la información de los paquetes de softwares usados en el trabajo para llevar a cabo la instalación del clúster.

Tabla 6
Softwares de Apertura

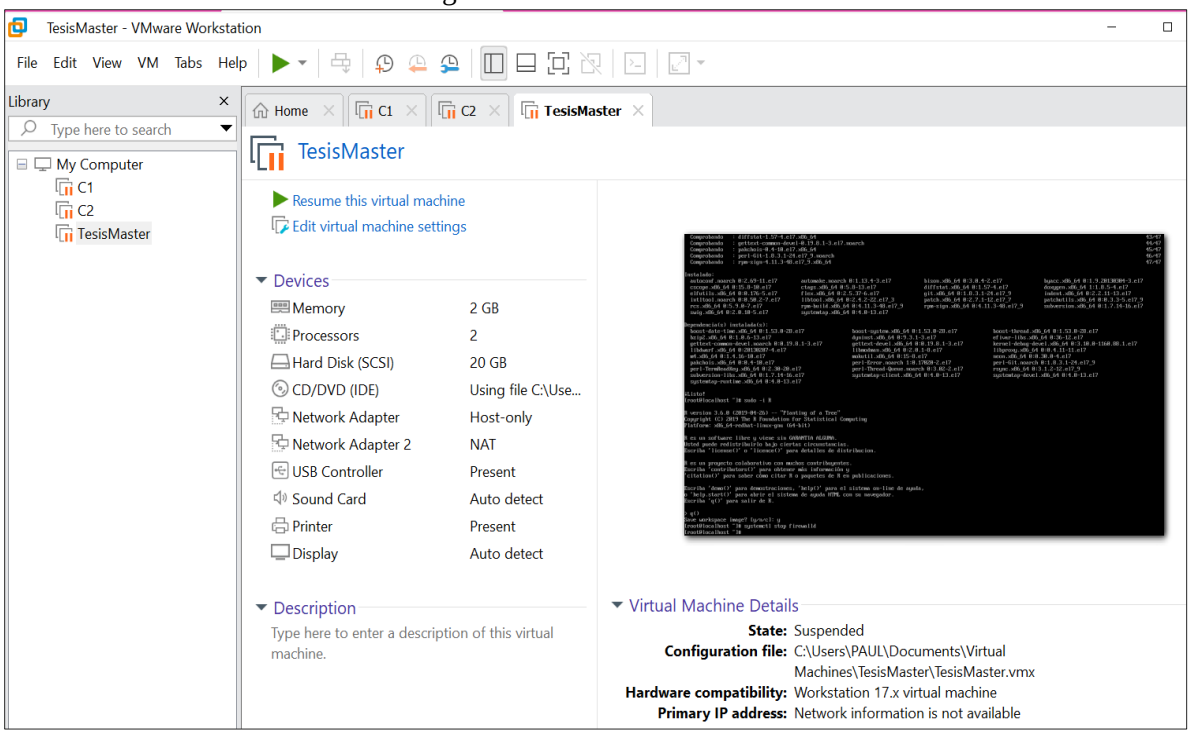
Software	Versión	Descripción
VMWare Workstation Pro	17x	Hypervisor para crear máquinas virtuales.
CentOS(x86_64)	7x	Sistema operativo de distribución de RHEL.
Yum	3.5.2	Gestor de descargas por paquetes para RHEL.

Fuente: El Autor

A continuación, se visualiza en el Gráfico 12 el clúster virtualizado con la herramienta VMWare Workstation Pro con cada uno de los nodos que conforman el Grid que son TesisMaster, C1 y C2 respectivamente. El proceso de instalación de las aplicaciones de apertura del clúster se encuentra

de manera detallada en el apartado de ANEXO 1.

Gráfico 12
Vista general del Cluster Virtualizado



Fuente: El Autor

Se presenta en la Tabla 7 las aplicaciones que servirán para el análisis, configuración y construcción del Grid TesisMaster, en el que posteriormente se realizará las pruebas de simulación de aproximación del *número Pi* en el entorno RStudio Server.

Tabla 7
Aplicaciones Operacionales para HPC

Software	Versión	Arquitectura	Descripción
OpenHPC	1.3.9.	X86_64	Paquetes de configuración para construir el cluster.
Ganglia Monitoring	3.7.2.	X86_64	Sistema de monitoreo de reporte estadístico del clúster.
Lenguaje R	3.5.	X86_64	Lenguaje de programación para cómputo estadístico y gráficos.
RStudio Server	1.0.136	X86_64	Entorno de desarrollo integrado para lenguaje R y ejecución del script.

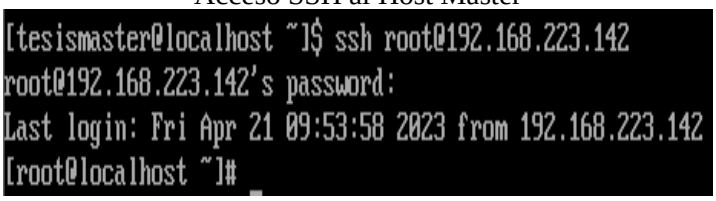
Fuente: El Autor

4.1.3. Instalación del Clúster OpenHPC

Se ingresa a la configuración interna del clúster mediante conexión *SSH (*por sus siglas en ingles *Secure Shell)* es un protocolo de seguridad para gestionar el control a través de internet por medio de acceso de autenticación. Para la instalación del clúster se accede con la dirección ip *192.168.223.142* que corresponde al nodo master donde se alojará todos los componentes de instalación y configuración como se observa en el Gráfico 13. El proceso completo de instalación y configuración del clúster se lo puede encontrar en el apartado de ANEXOS 2,3 y 4.

Se accede al ip pública por SSH para realizar la configuración
[tesismaster@localhost ~]# ssh root@192.168.223.142

Gráfico 13
Acceso SSH al Host Master

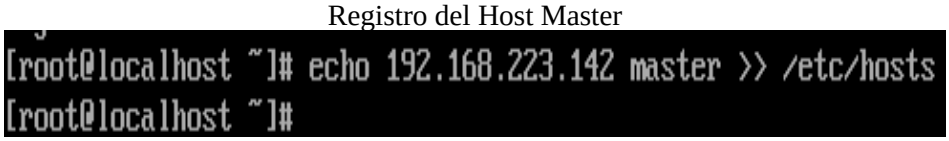


Fuente: El Autor

Se puede definir las direcciones ip de los nodos mediante el comando ***"echo"*** o ingresando al directorio de hosts ***"/etc/hosts"*** a través del comando **"vi /etc/hosts"**. El mismo proceso se lo realiza para la configuración en el nodo master y los nodos cómputos C1 y C2 que se visualizan en los Gráficos 14 y 15 respectivamente.

Se asigna el nombre del host "master" a la red local
[root@localhost ~]# echo 192.168.223.142 master >> /etc/hosts

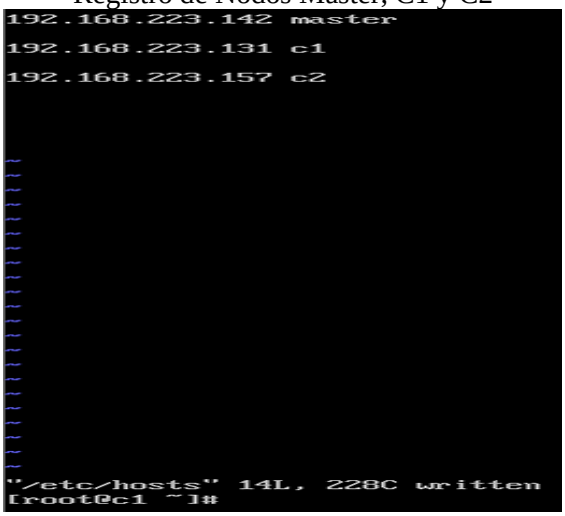
Gráfico 14
Registro del Host Master



Fuente: El Autor

Se accede al directorio de host "master" para registrar los nodos cómputos
[root@localhost ~]# vi /etc/hosts

Gráfico 15
Registro de Nodos Master, C1 y C2



Fuente: El Autor

Se procede a *deshabilitar el firewall* porque puede ocasionar conflicto en la instalación de los componentes de OpenHPC mediante los siguientes comandos que se muestra en el siguiente Gráfico 16. Este proceso se lo debe realizar también en los nodos cómputos C1 y C2.

Se deshabilita el firewall del SO
[root@localhost ~]# systemctl disable firewalld
[root@localhost ~]# systemctl stop firewalld

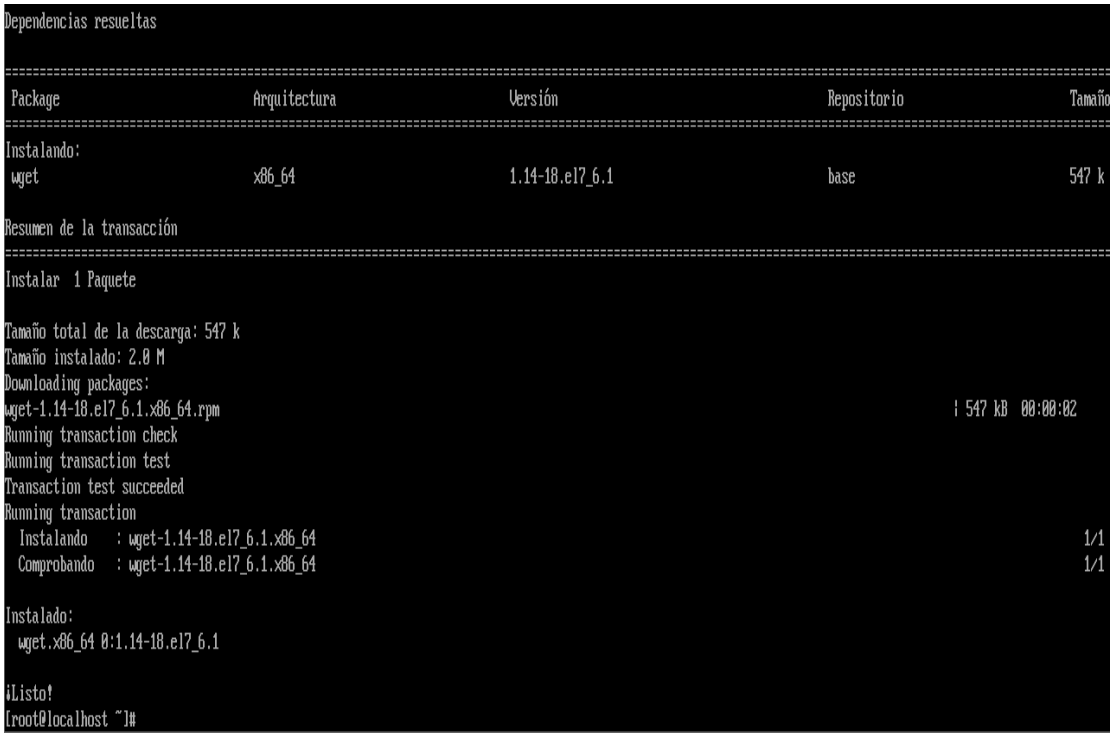
Gráfico 16
Desactivación del Firewall

Fuente: El Autor

Se recomienda actualizar el sistema operativo de CentOS 7 en los nodos del grid para obtener nuevas actualizaciones referente a la instalación del clúster que se aprecia en Gráfico 17. Lo mismo se debe realizar para todos los nodos master, C1 y C2.

Se actualiza el Sistema Operativo CentOS 7
[root@localhost ~]# yum -y update
[root@localhost ~]# yum -y install wget

Gráfico 17
Actualización del SO CentOS 7



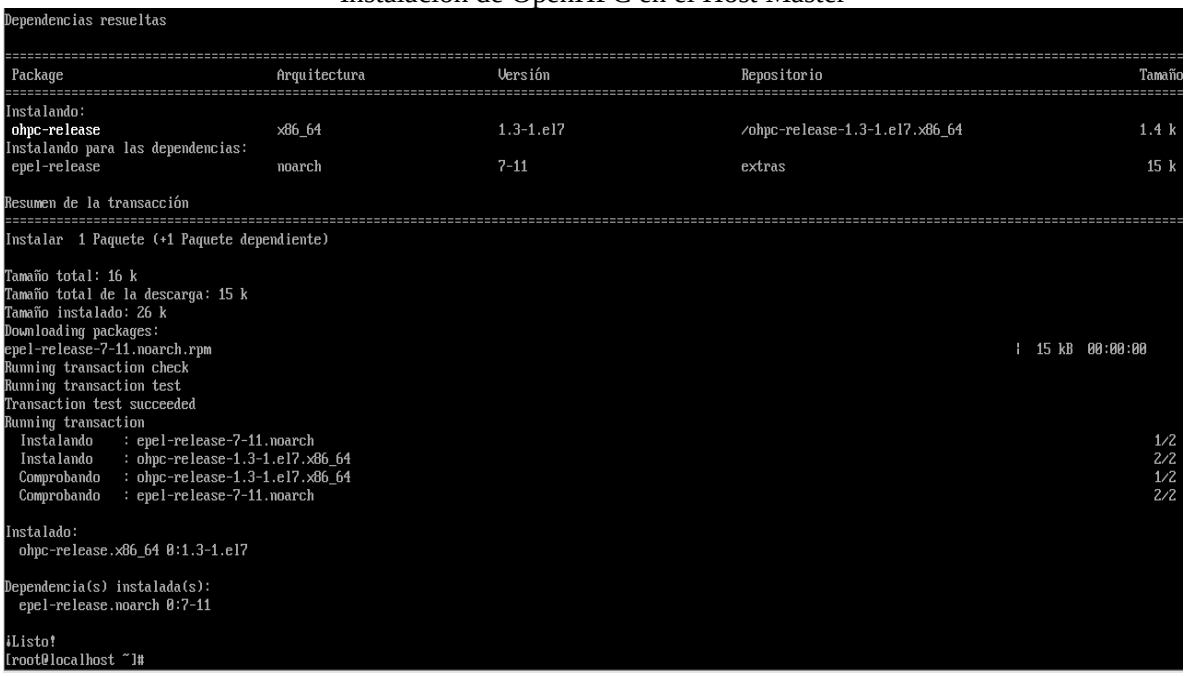
Fuente: El Autor

Se procede con la instalación del clúster OpenHPC en el nodo master mediante el comando ***"yum"*** (por sus siglas en ingles yellowdog updater modified) para instalar distribución de paquetes en Linux que se visualiza en el siguiente Gráfico 18.

Se instala OpenHPC en el nodo master
[root@localhost ~]# yum -y install
http://build.openhpc.community/OpenHPC:/1.3/CentOS_7/x86_64/ohpc-release-1.31.el7.x86_64.rpm

Gráfico 18

Instalación de OpenHPC en el Host Master

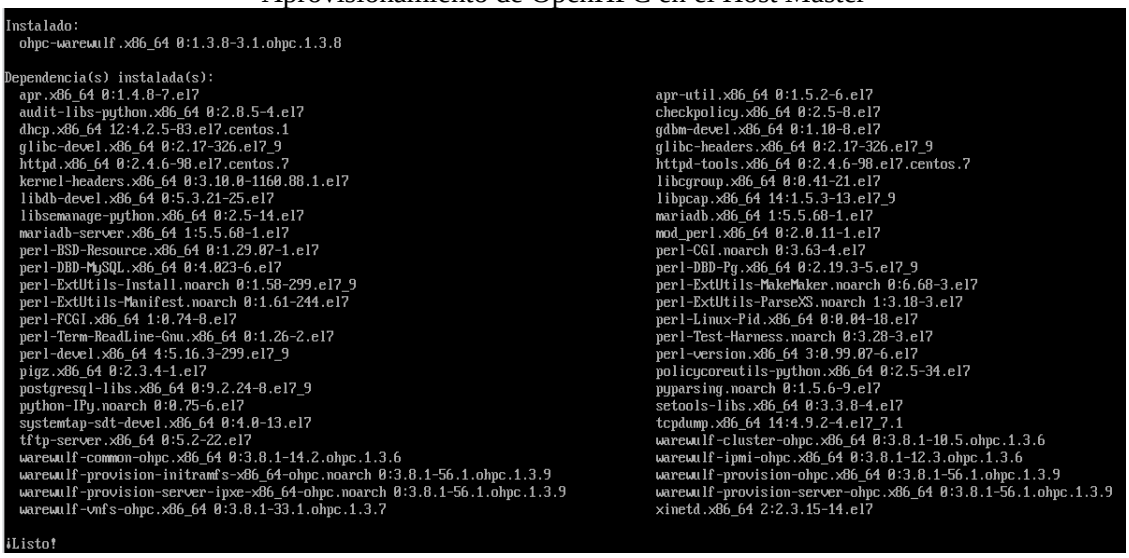


Fuente: El Autor

Se instala los servicios de aprovisionamiento en el nodo master porque provee una gama de componentes para la configuración del clúster que se muestra en el siguiente Gráfico 19.

Se instala el aprovisionamiento en el nodo master
[root@localhost ~]# yum -y install ohpc-base
[root@localhost ~]# yum -y install ohpc-warewulf

Gráfico 19
Aprovisionamiento de OpenHPC en el Host Master



Fuente: El Autor

Se procede con la instalación de sincronizar los relojes de la hora local con la hora global en el host master mediante el protocolo *NTPD* (por sus siglas en ingles Network Time Protocol Date) con el servidor de ***"time.haii.or.th"*** que servirá para el monitoreo del clúster en tiempo real como se muestra en el siguiente Gráfico 20.

Se habilita el servicio NTP en el nodo master
[root@localhost ~]# systemctl enable ntpd.service
[root@localhost ~]# echo "server time.haii.or.th" >> /etc/ntp.conf
[root@localhost ~]# systemctl restart ntpd

Gráfico 20
Habilitación del Servicio NTP

Fuente: El Autor

Se instala los servicios de gestión de recursos en el nodo master con el servidor de gestor de cargas de trabajo Slurm para luego añadir una imagen de cómputo que se observa en el siguiente Gráfico

21.

Se añade los servicios de gestión de recursos en el nodo master.
[root@localhost ~]# yum -y install ohpc-slurm-server
[root@localhost ~]# perl -pi -e "s/ControlMachine=\S+/ControlMachine=master/" /etc/slurm/slurm.conf

Gráfico 21
Gestión de Recursos Host Master

Fuente: El Autor

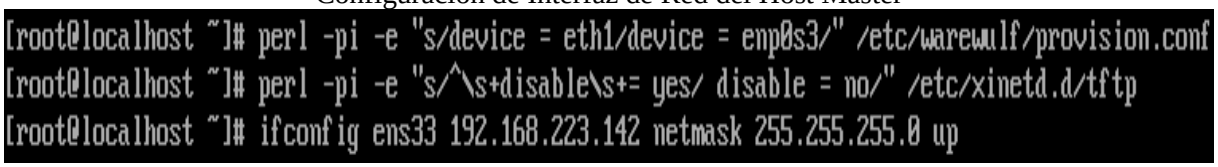
Se configura la interfaz de red interna en el nodo master y habilitarla para su aprovisionamiento a los demás nodos como se detalla en el siguiente Gráfico 22.

Configuración de interfaz interna.
[root@localhost ~]# perl -pi -e "s/device = eth1/device = ens33/" /etc/warewulf/provision.conf

Habilitar el servicio tftp para la distribución de imagen en el nodo de cómputo.
[root@localhost ~]# perl -pi -e "s/^\s+disable\s+= yes/ disable = no/" /etc/xinetd.d/tftp

Habilitar la interfaz interna para aprovisionamiento.
[root@localhost ~]# ifconfig ens33 192.168.223.142 netmask 255.255.255.0 up

Gráfico 22
Configuración de Interfaz de Red del Host Master



Fuente: El Autor

Se habilita los servicios de aprovisionamiento de la base de datos MariaDB, servidor web http y

el protocolo *DHCP* (por sus siglas en ingles Dynamic Host Configuration Protocol) que se puede apreciar en el siguiente Gráfico 23.

Gráfico 23
Habilitación de Aprovisionamiento

```
[root@localhost ~]# systemctl restart xinetd
[root@localhost ~]# systemctl enable mariadb.service
[root@localhost ~]# systemctl restart mariadb
[root@localhost ~]# systemctl enable httpd.service
[root@localhost ~]# systemctl restart httpd
[root@localhost ~]# systemctl enable dhcpd.service
Created symlink from /etc/systemd/system/multi-user.target.wants/dhcpd.service to /usr/lib/systemd/system/dhcpd.service.
[root@localhost ~]#
```

Fuente: El Autor

Se inicializa los servicios de la base de datos de MariaDB con sus credenciales de acceso *ssh* por Warewulf para la muestra de datos del clúster en la plataforma HPC que se puede apreciar en el siguiente Gráfico 24.

Se inicializa base de datos Warewulf y accesos ssh.
[root@localhost ~]# wwinit database [root@localhost ~]# wwinit ssh_keys

Gráfico 24
Iniciación de BD MariaDB

```
¡Listo!
[root@localhost ~]# wwinit database
database:     Checking to see if RPM 'mysql-server' is installed                NO
database:     Checking to see if RPM 'mariadb-server' is installed              OK
database:     Activating Systemd unit: mariadb
database:      + /bin/systemctl -q enable mariadb.service                       OK
database:      + /bin/systemctl -q restart mariadb.service                      OK
database:      + mysqladmin --defaults-extra-file=/tmp/0.RPeNMseD5wZX/my.cnf OK
database:     Database version: UNDEF (need to create database)
database:      + mysql --defaults-extra-file=/tmp/0.RPeNMseD5wZX/my.cnf ware OK
database:      + mysql --defaults-extra-file=/tmp/0.RPeNMseD5wZX/my.cnf ware OK
database:      + mysql --defaults-extra-file=/tmp/0.RPeNMseD5wZX/my.cnf ware OK
database:     Checking binstore kind                                          SUCCESS
Done.
[root@localhost ~]# wwinit ssh_keys
ssh_keys:     Checking ssh keys for root                                        OK
ssh_keys:     Checking root's ssh config                                        OK
ssh_keys:     Checking for default RSA host key for nodes                       NO
ssh_keys:     Creating default node ssh_host_rsa_key:
ssh_keys:      + ssh-keygen -q -t rsa -f /etc/warewulf/vnfs/ssh/ssh_host_rsa OK
ssh_keys:     Checking for default DSA host key for nodes                       NO
ssh_keys:     Creating default node ssh_host_dsa_key:
ssh_keys:      + ssh-keygen -q -t dsa -f /etc/warewulf/vnfs/ssh/ssh_host_dsa OK
ssh_keys:     Checking for default ECDSA host key for nodes                     NO
ssh_keys:     Creating default node ssh_host_ecdsa_key:
                                                                                OK
ssh_keys:     Checking for default Ed25519 host key for nodes                   NO
ssh_keys:     Creating default node ssh_host_ed25519_key:
                                                                                OK
Done.
[root@localhost ~]#
```

Fuente: El Autor

Se añade el servidor NTP en el nodo master para obtener información en tiempo real del clúster OpenHPC que se puede visualizar en el siguiente Gráfico 25.

Se añade servidor NTP del nodo master hacia los nodos cómputos
[root@localhost ~]# echo "server 192.168.223.142" >> $CHROOT/etc/ntp.conf
[root@localhost ~]# chroot $CHROOT systemctl enable ntpd

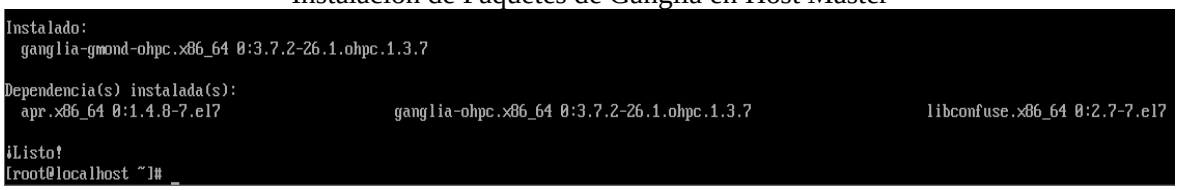
Gráfico 25
Adición de Servidor NTP en Host Master

Fuente: El Autor

Se realiza la instalación del paquete de servicios de Ganglia en el host master que monitorea el desempeño del grid en el clúster OpenHPC y que se aprecia en el siguiente Gráfico 26.

Se instala Ganglia meta-package en el nodo master
[root@localhost ~]# yum -y install ohpc-ganglia
Se instala Ganglia en los nodos computo
[root@localhost ~]# yum -y --installroot=$CHROOT install ganglia-gmond-ohpc

Gráfico 26
Instalación de Paquetes de Ganglia en Host Master



Fuente: El Autor

Se instala el demonio “gmond” de ganglia en el host master y los nodos computos C1 y C2 que se visualiza en el siguiente Gráfico 27.

Gráfico 27
Instalación Demonios de Ganglia en Host Master

Fuente: El Autor

Se configura el demonio gmond de ganglia en el host master, se añade el nombre del grid que es del *TesisMaster* y se habilita los servicios de ganglia para ingresar al grid desde el navegador web en la dirección https://localhost.com/ganglia que se aprecia en el siguiente Gráfico 28.

Uso de script de configuración para habilitar receptor único del nodo master
[root@localhost ~]# cp /opt/ohpc/pub/examples/ganglia/gmond.conf /etc/ganglia/gmond.conf
[root@localhost ~]# perl -pi -e "s/<sms>/master/" /etc/ganglia/gmond.conf
Se añade configuración al nodo cómputo y se coloca el nombre del grid
[root@localhost ~]# cp /etc/ganglia/gmond.conf $CHROOT/etc/ganglia/gmond.conf
[root@localhost ~]# echo "gridname TesisMaster..." >> /etc/ganglia/gmetad.conf
Se comienza habilitar los servicios de ganglia
[root@localhost ~]# systemctl enable gmond
[root@localhost ~]# systemctl enable gmetad
[root@localhost ~]# systemctl start gmond
[root@localhost ~]# systemctl start gmetad
[root@localhost ~]# chroot $CHROOT systemctl enable gmond
Se restaura el servidor web
[root@localhost ~]# systemctl try-restart httpd

Gráfico 28
Habilitación de los Servicios de Ganglia en Grid TesisMaster

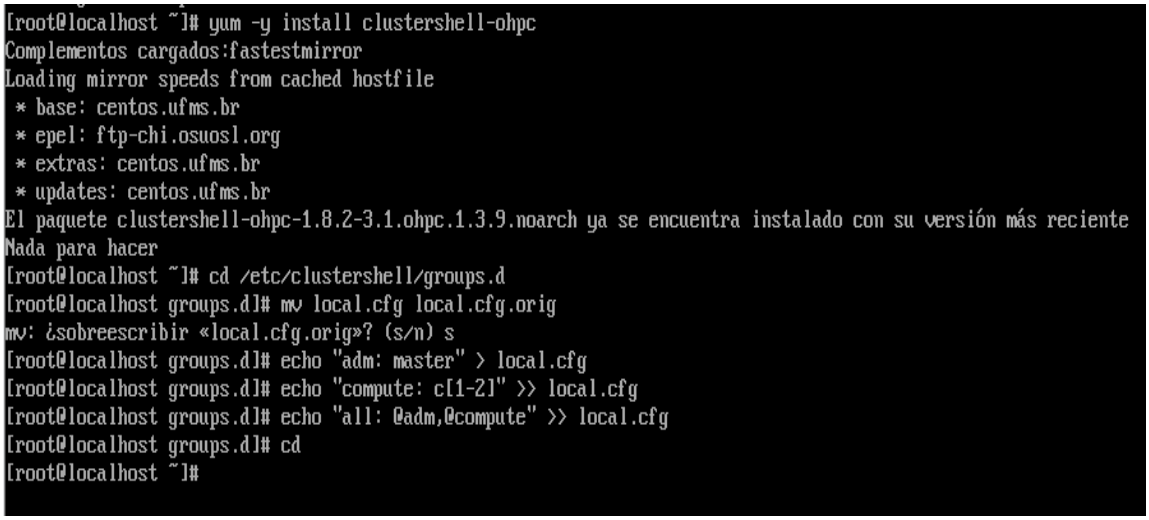
Fuente: El Autor

Se instala un *ClusterShell* basado en librería de Python para ejecutar los comandos en paralelo en los nodos del clúster. Esta configuración define el grid compuesto por dos nodos cómputos como se puede visualizar en el siguiente Gráfico 29.

Se instala ClusterShell
[root@localhost ~]# yum -y install clustershell-ohpc
Se configura la definición del nodo
[root@localhost ~]# cd /etc/clustershell/groups.d
[root@localhost ~]# mv local.cfg local.cfg.orig
[root@localhost ~]# echo "adm: master" > local.cfg
[root@localhost ~]# echo "compute: c [1-2]" >> local.cfg
[root@localhost ~]# echo "all: @adm,@compute" >> local.cfg

Gráfico 29

Instalación de Clustershell en Host Master



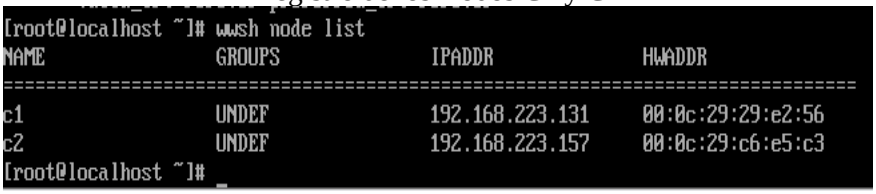
Fuente: El Autor

Mediante el comando ***"wwsh node list"*** se muestra la información de la dirección IP y MAC de los nodos cómputo C1 y C2 en el host master que conforman el grid *TesisMaster* que se observa en el siguiente Gráfico 30.

Se añaden los nodos al almacenamiento de datos Warewulf.
[root@localhost ~]# wwsh -y node new c1 --ipaddr=192.168.23.131 --hwaddr=00:0C:29:29:E2:56 -D ens36
[root@localhost ~]# wwsh -y node new c2 --ipaddr=192.168.223.157 --hwaddr=00:0C:29:C6:E5:C3 -D ens36
[root@localhost ~]# wwsh node list

Gráfico 30

Registro de los Nodos C1 y C2

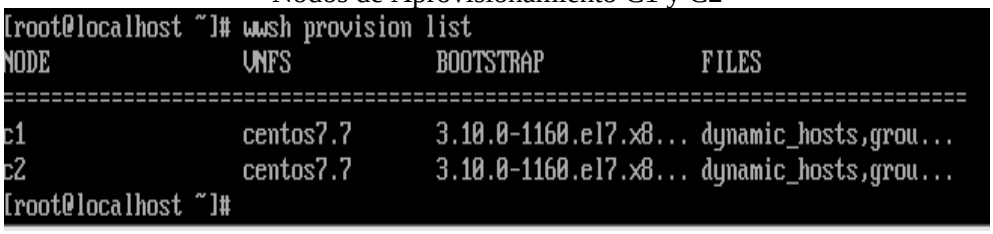


Fuente: El Autor

Se define la imagen de *VNFS* sistema de archivo de nodo virtual (por sus siglas en ingles Virtual Node File System) que restaura los servicios de ganglia y dhcp para visualizar el grid en el navegador web como se aprecia en el siguiente Gráfico 31.

Se define la imagen VNFS.
[root@localhost ~]# wwsh -y provision set "c1" --vnfs=centos7.7 --bootstrap=`uname -r` --files=dynamic_hosts,passwd,group,shadow,slurm.conf,munge.key,network
[root@localhost ~]# wwsh -y provision set "c2" --vnfs=centos7.7 --bootstrap=`uname -r` --files=dynamic_hosts,passwd,group,shadow,slurm.conf,munge.key,network
Se restaura los servicios de ganglia/dhcp.
[root@localhost ~]# systemctl restart gmond
[root@localhost ~]# systemctl restart gmetad
[root@localhost ~]# systemctl restart dhcpd

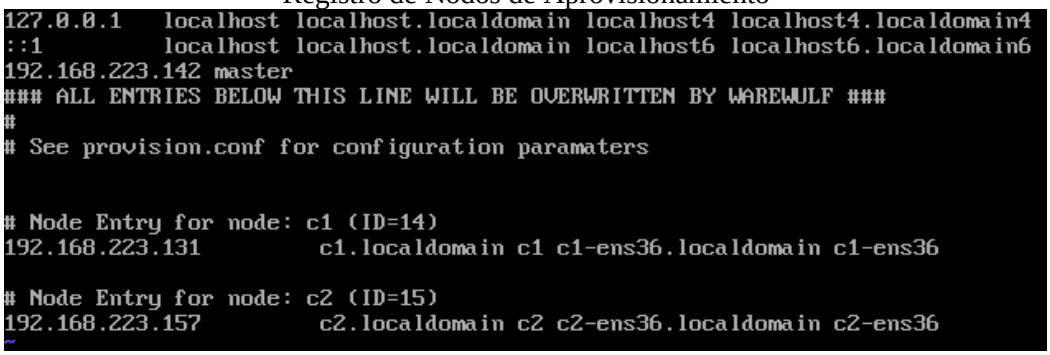
Gráfico 31
Nodos de Aprovisionamiento C1 y C2



Fuente: El Autor

Se verifica en el siguiente Gráfico 32 los servicios de aprovisionamientos de los nodos cómputo registrados en el directorio ***"/etc/hosts"*** que se encuentran en el host master.

Gráfico 32
Registro de Nodos de Aprovisionamiento

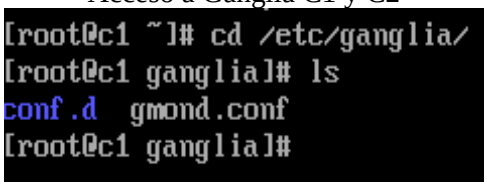


Fuente: El Autor

En los nodos C1 y C2 se ingresa al directorio de ganglia para luego acceder a la configuración del demonio gmond que se aprecia en el siguiente Gráfico 33.

Acceso al directorio de ganglia.
[root@c1 ~]# cd /etc/ganglia/

Gráfico 33
Acceso a Ganglia C1 y C2

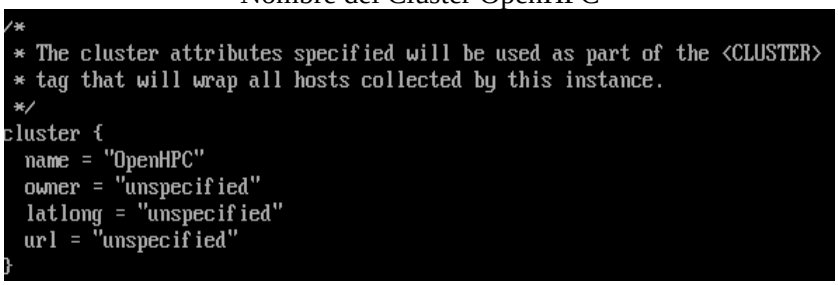


Fuente: El Autor

Se realiza el ingreso a la configuración del demonio gmond en los nodos C1 y C2 para efectuar los cambios en los parámetros en el nombre del clúster que en este caso es el de "***OpenHPC"***, el *udp_send_channel*, *udp_recv_channel* y *tcp_accept_channel*. Debe quedar configurado tal como se muestra en los siguientes Gráficos 34, 35 y 36. Al finalizar se digita el comando :wq para guardar los cambios respectivos.

Acceso a la configuración del demonio gmond.
[root@c1 ganglia]# vi gmond.conf

Gráfico 34
Nombre del Clúster OpenHPC

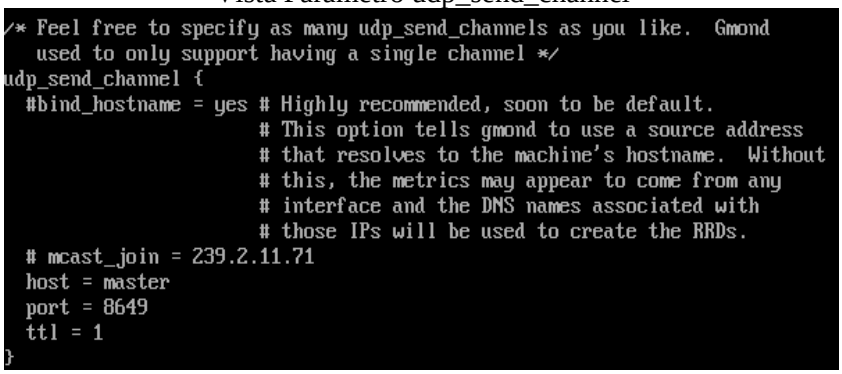


Fuente: El Autor

En los nodos C1 y C2 se comenta con el signo # los parámetros de *bind_hostname* y *mcast_join*. Donde sólo quede habilitado los parámetros de *host, port y ttl* que se observa en el siguiente Gráfico 35.

Gráfico 35

Vista Parámetro udp_send_channel

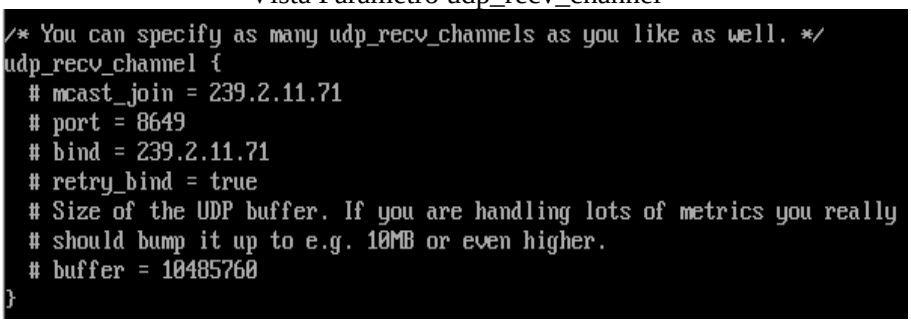


Fuente: El Autor

Se digita con el signo # los parámetros de *mcast_join, bind, port, retry_bind*, etc en los nodos C1 y C2 que se visualiza en el siguiente Gráfico 36.

Gráfico 36

Vista Parámetro udp_recv_channel

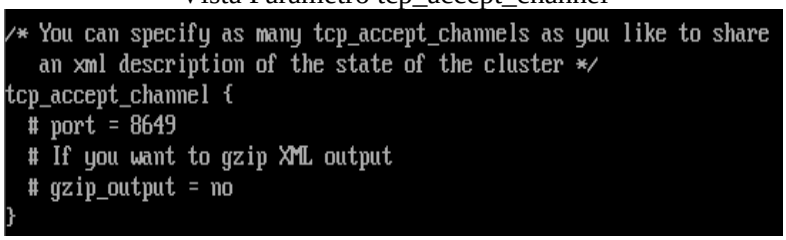


Fuente: El Autor

Se aprecia en el siguiente Gráfico 37 como debe quedar digitado con el signo # los parámetros de *port, gzip XML y gzip_output*.

Gráfico 37

Vista Parámetro tcp_accept_channel

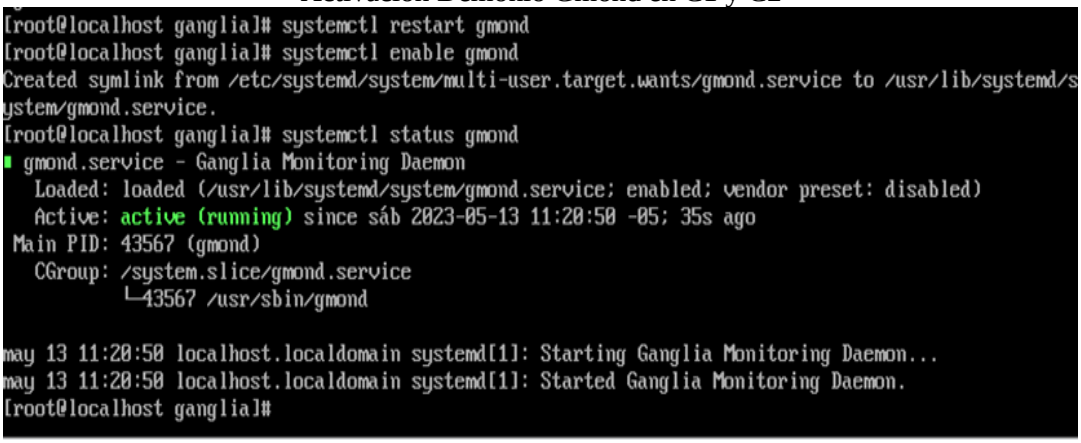


Fuente: El Autor

Se observa en el siguiente Gráfico 38 la habilitación del demonio gmond en los nodos C1 y C2 con estado ***activo*** y pueda ser ejecutado de manera correcta en el navegador web.

Se habilita el demonio gmond.
[root@c1 ganglia]# systemctl restart gmond
[root@c1 ganglia]# systemctl enable gmond
[root@c1 ganglia]# systemctl status gmond

Gráfico 38
Activación Demonio Gmond en C1 y C2



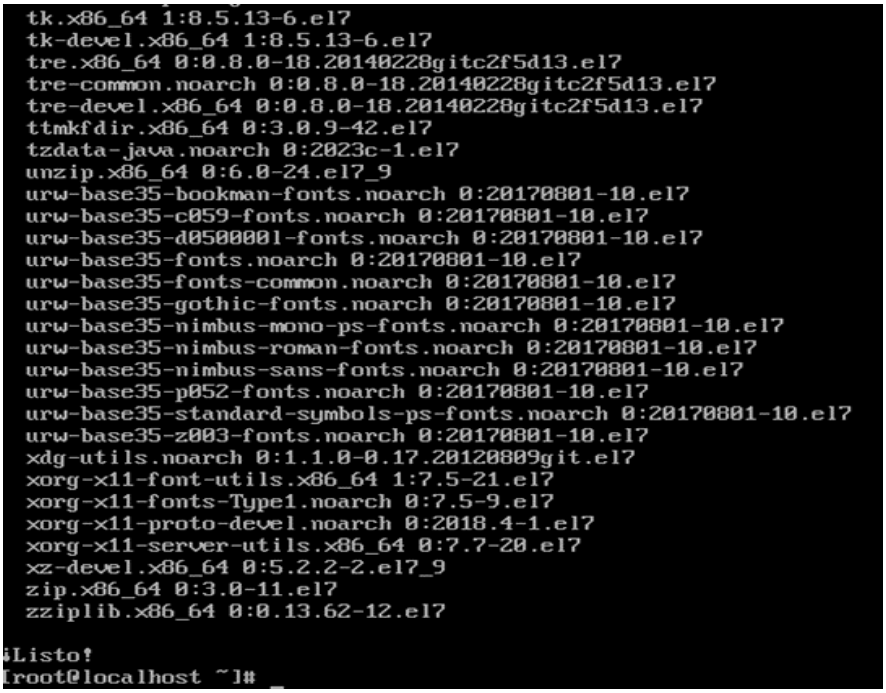
Fuente: El Autor

Se realiza la instalación del lenguaje R en el host master que se muestra a continuación en el siguiente Gráfico 39.

Se instala el lenguaje R en el nodo master.
[root@localhost ~]# yum install R -y

Gráfico 39

Instalación de R en Host Master



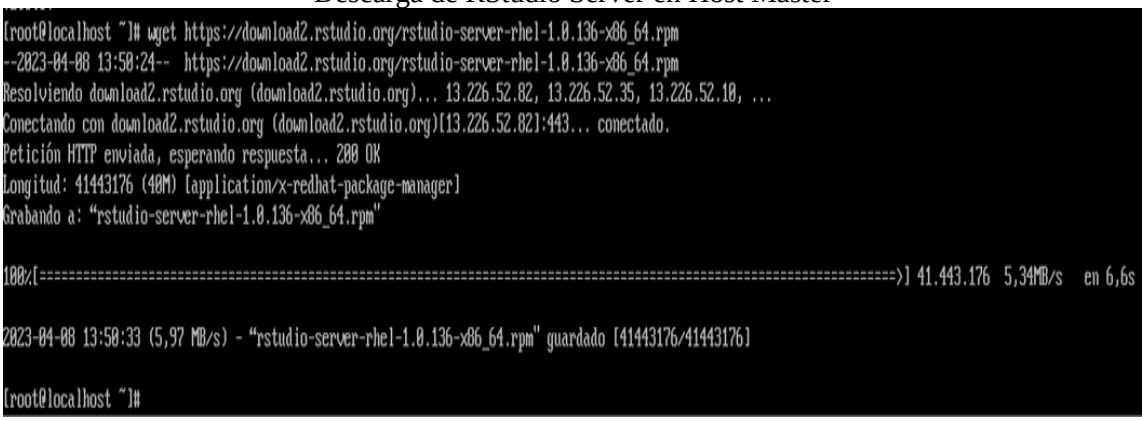
Fuente: El Autor

Se procede con la descarga de RStudio Server en el host master con el siguiente comando que se visualiza a continuación en el siguiente Gráfico 40.

Se descarga el entorno RStudio Server en el nodo master.
[root@localhost ~]# wget https://download2.rstudio.org/rstudio-server-rhel-1.0.136-x86_64.rpm

Gráfico 40

Descarga de RStudio Server en Host Master



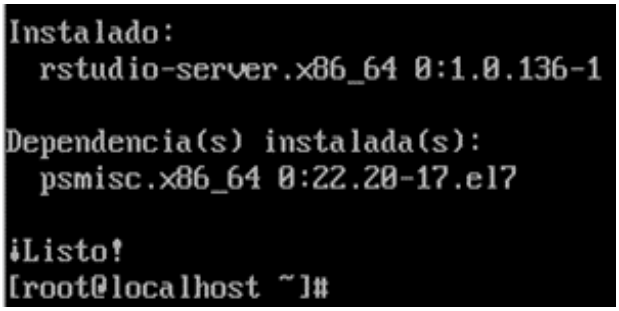
Fuente: El Autor

Se realiza la instalación de RStudio Server en el nodo master con el correspondiente comando que

se muestra a continuación en el siguiente Gráfico 41.

Se instala el entorno RStudio Server en el nodo master.
[root@localhost ~]# yum install –nogpgcheck rstudio-server-rhel-1.0.136-x86_64.rpm -y

Gráfico 41
Instalación de RStudio Server en Host Master



Fuente: El Autor

Se comprueba que se encuentre en estado activo el servicio de RStudio Server para ser ejecutado en el host master mediante el comando que se refleja en el siguiente Gráfico 42.

Se comprueba RStudio Server.
[root@localhost ~]# systemctl status rstudio-server.service

Gráfico 42
Estado Activo de RStudio Server

Fuente: El Autor

4.1.4. Script de aproximación del número Pi

El script fue facilitado por la División Estadística del Departamento de Población y Ciencias de la

Salud Pública (USC Biostatistics) de la Universidad del Sureste de California disponible en su repositorio de GitHub https://github.com/USCbiostats/slurmr-workshop. Utiliza una versión paralela de mclapply que devuelve una lista de la misma longitud, donde cada elemento es el resultado correspondiente de la simulación. (Yon, 2021). Se presenta a continuación en la Tabla 8 el script del *número Pi* desarrollado en lenguaje R donde muestra sus parámetros de modelos, la función a simular y los métodos de aproximación con el uso de *mclapply*.

Tabla 8

Script Aproximación *Número Pi*

```
02-mclapply.R
# Model parameters
nsims  <- 1e3
n      <- 1e4
ncores <- 4L

# Function to simulate pi
simpi <- function(i) {

  p <- matrix(runif(n*2, -1, 1), ncol = 2)
  mean(sqrt(rowSums(p^2)) <= 1) * 4

}

# Approximation
set.seed(12322)
ans <- parallel::mclapply(1:nsims, simpi, mc.cores = ncores)
ans <- unlist(ans)

message("Pi: ", mean(ans))

saveRDS(ans, "02-mclpply.rds")
```

Fuente: (USCbiostats, 2021)

4.2. Presentación de la Plataforma HPC y entorno del script del *número Pi*

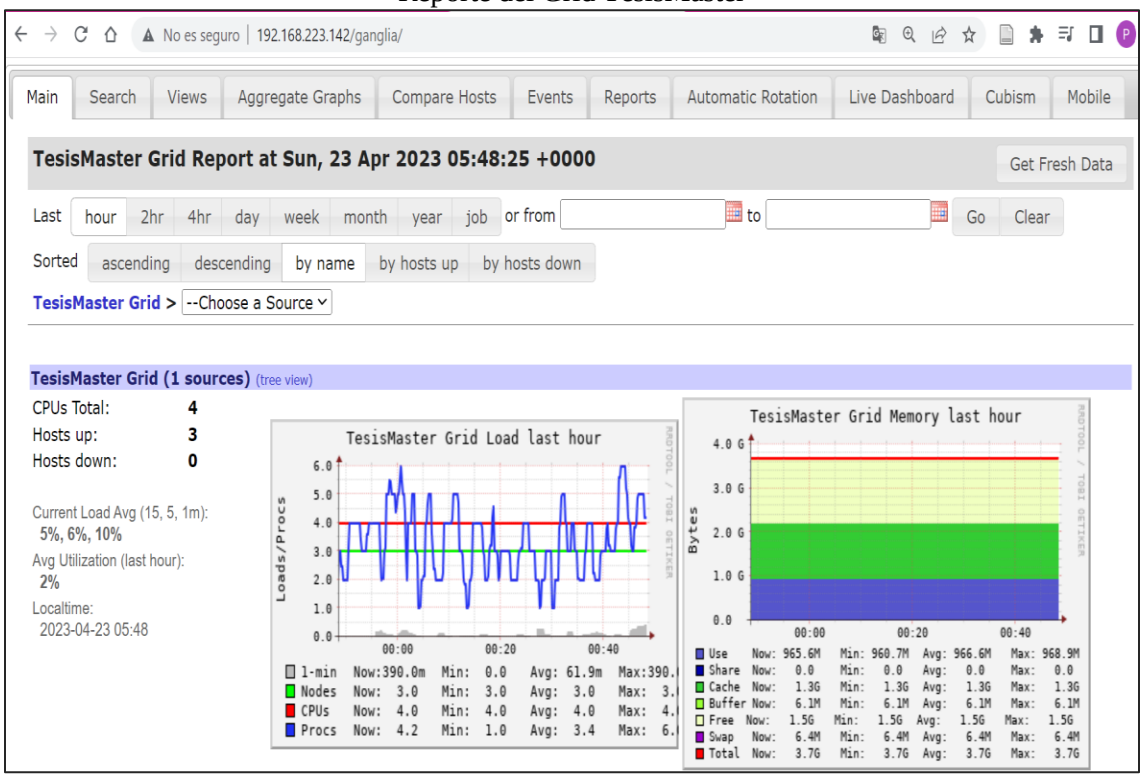
Se presenta el clúster OpenHPC con el grid TesisMaster mediante la aplicación de Ganglia para monitorear su estado de desempeño. Se procede con la ejecución del script de aproximación del *número Pi* en el entorno RStudio Server, donde se verifica los resultados de su simulación y se realiza el análisis de la plataforma HPC.

4.2.1. Ganglia Monitoring para OpenHPC

Ganglia Monitoring es una herramienta que monitorea el estado de desempeño de un clúster HPC

compuesto por cada uno de los nodos que muestra el comportamiento del grid instalado en el nodo master. Se divisa en el siguiente Gráfico 43 el entorno general de Ganglia del grid TesisMaster con sus métricas de carga de trabajo y red.

Gráfico 43
Reporte del Grid TesisMaster

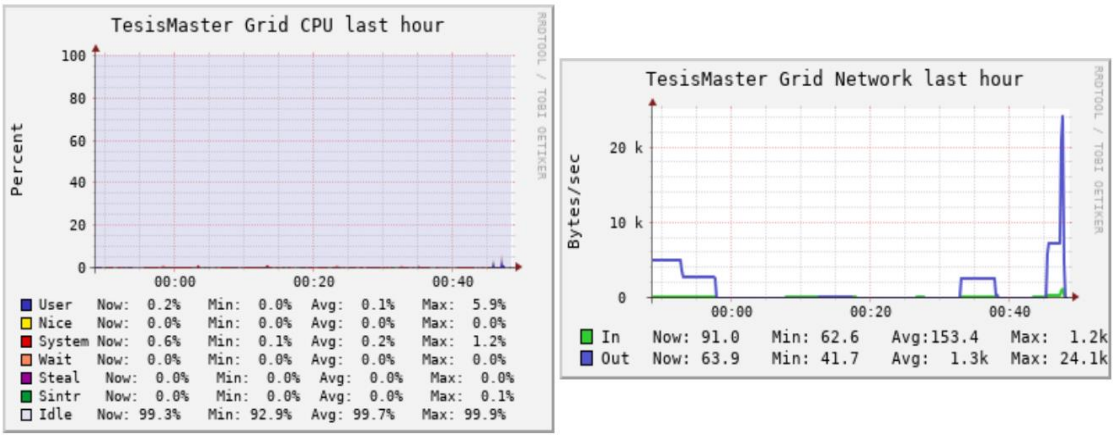


Fuente: El Autor

Se visualiza en el Gráfico 44 el comportamiento de las métricas de reportes del CPU y envío de datos en la red del grid TesisMaster.

Gráfico 44

Reporte de CPU y Red

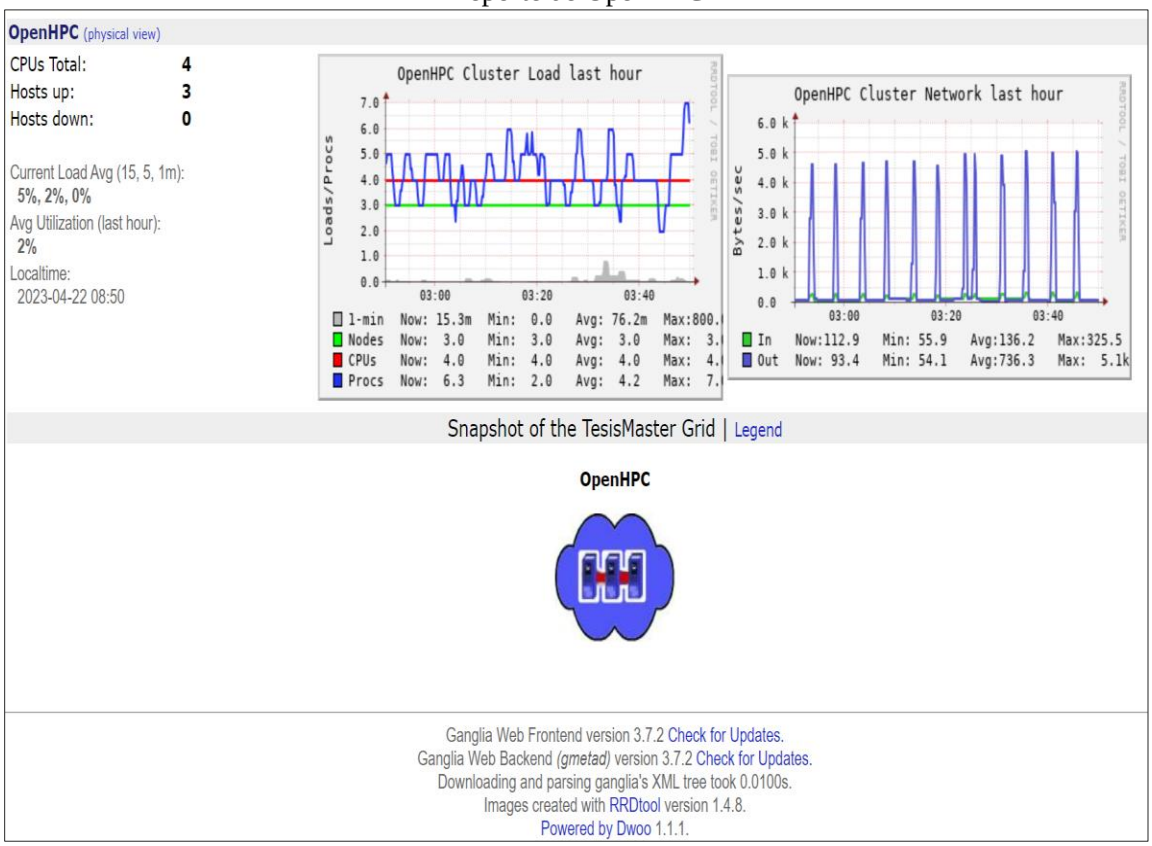


Fuente: El Autor

Posteriormente se aprecia en el Gráfico 45 el reporte de carga de trabajo y tráfico de red del clúster físico de OpenHPC.

Gráfico 45

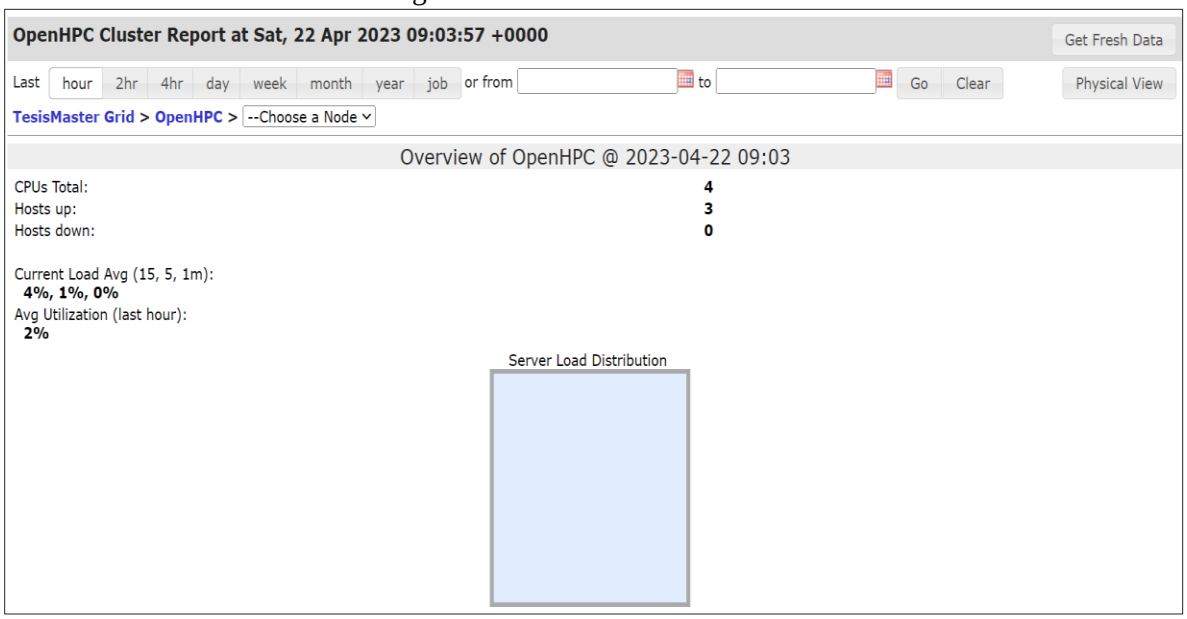
Reporte de OpenHPC



Fuente: El Autor

Se aprecia en el siguiente Gráfico 46 la distribución de carga del servidor del grid TesisMaster en el clúster OpenHPC donde muestra su porcentaje de promedio de carga actual y de utilización.

Gráfico 46
Carga Promedio del Grid TesisMaster

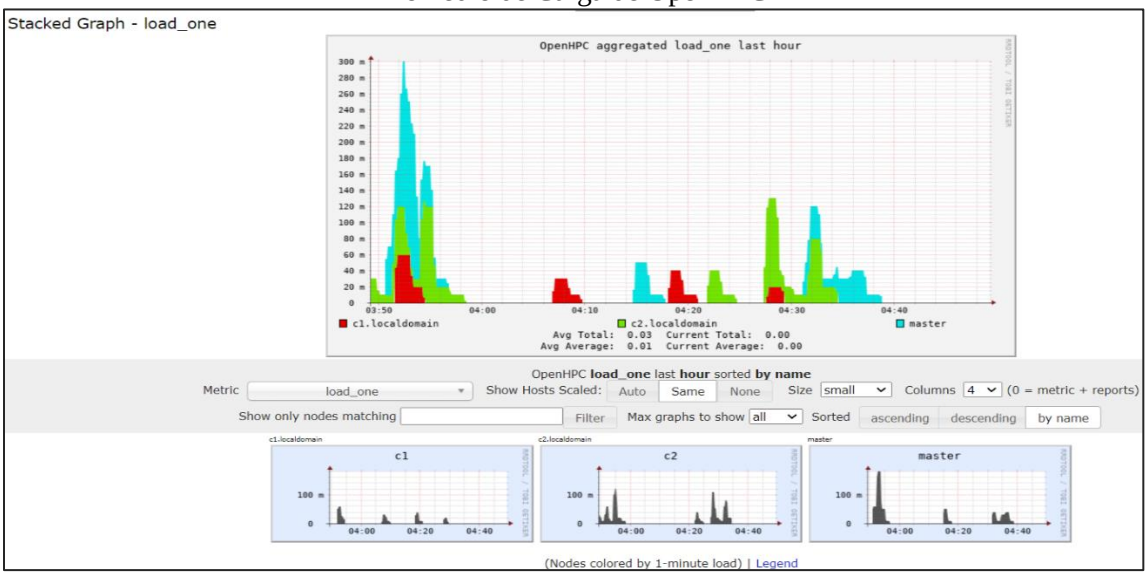


Fuente: El Autor

Se presenta un resumen sobre el consumo de carga única del clúster OpenHPC para el grid TesisMaster conformados por los nodos master, C1 y C2 que se visualizan en el siguiente Gráfico 47.

Gráfico 47

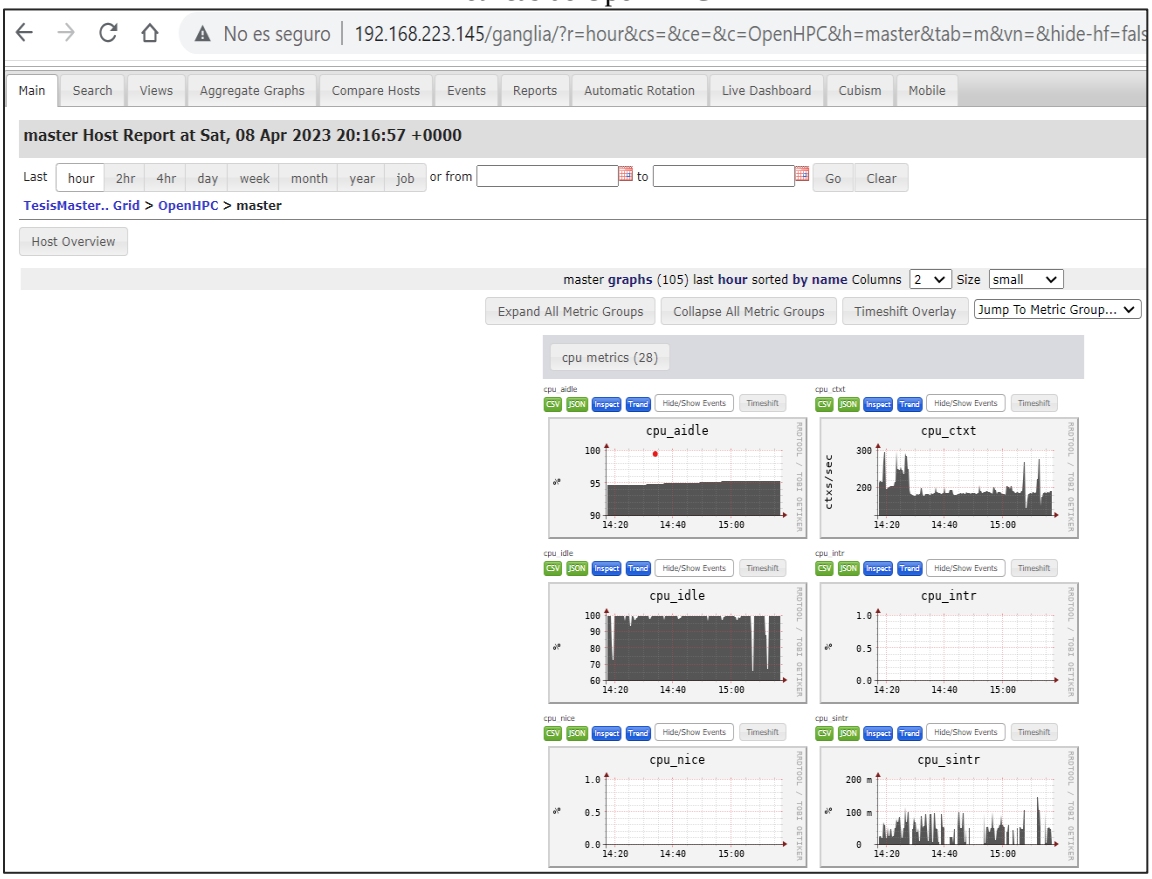
Promedio de Carga de OpenHPC



Fuente: El Autor

Por último, Se muestra en el Gráfico 48 un resumen de las métricas de procesos del grid TesisMaster para el clúster físico OpenHPC en el nodo master, lo que servirá como base para el análisis de la plataforma HPC.

Gráfico 48
Métricas de Open HPC



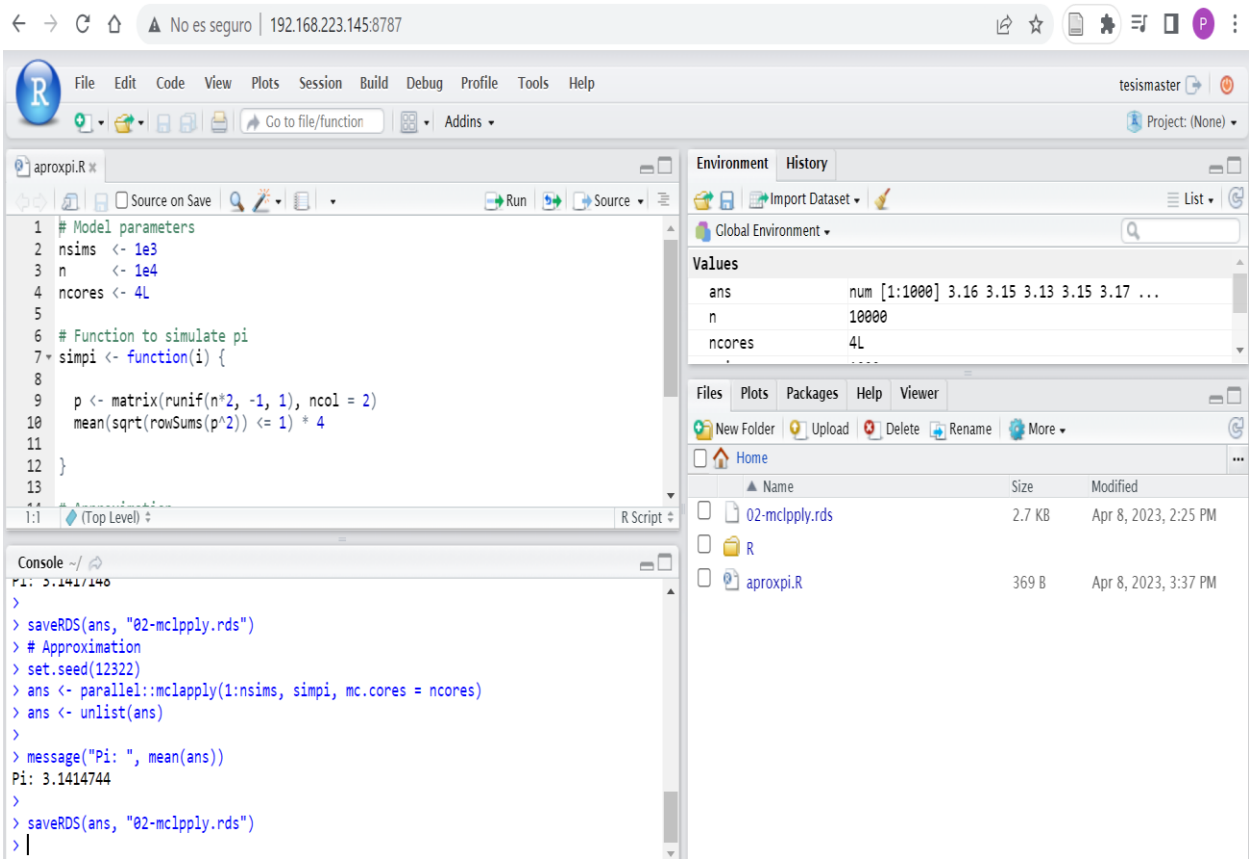
Fuente: El Autor

4.2.2. Script de aproximación del número pi en RStudio Server

Se realizó la instrucción de copiar-pegar del script de aproximación del *número Pi* en el entorno RStudio Server para luego proceder con las ejecuciones y obtener los resultados de las pruebas de la simulación. El procedimiento de instalación y configuración del entorno RStudio Server en el nodo master se lo encuentra en el apartado de ANEXO 4.

Se visualiza en el Gráfico 49 el entorno general de RStudio Server que está compuesto básicamente de un entorno global de desarrollo, gestor de archivos y consola.

Gráfico 49
Entorno RStudio Server



Fuente: El Autor

4.3. Análisis de Desempeño y Pruebas de la Plataforma HPC

En el análisis de desempeno se realiza la ejecución del script de aproximación del *número Pi* en el entorno RStudio Server y se procede con el análisis del clúster OpenHPC en la plataforma web de Ganglia Monitoring que proporciona un resumen general de los nodos que conforman el grid TesisMaster.

4.3.1. Ejecución del Script del número Pi

La ejecución del script de aproximación del *número Pi* se puede realizar en dos entornos como el de RStudio Server o en la terminal de CentOS. Para llevar a cabo la simulación se ejecuta la librería

paralela mclapply en los entornos planteados. Se muestra en la siguiente Tabla 9 el método de la librería mclapply que presenta el resultado aproximado del *número Pi*.

Tabla 9

Librería Paralela Mclapply

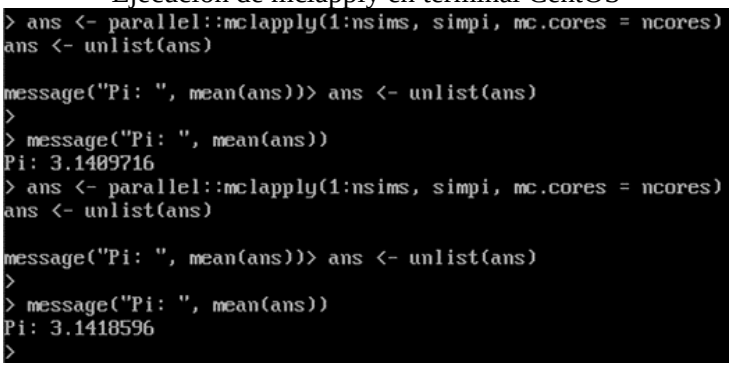
```
ans <- parallel::mclapply(1:nsims, simpi, mc.cores = ncores)
ans <- unlist(ans)
message("Pi: ", mean(ans))
```

Fuente: (USCbiostats, 2021)

Se visualiza en el siguiente Gráfico 50 la ejecución de la librería paralela mclapply en la terminal del nodo master que se observa el resultado de dos ejecuciones para la simulación del aproximado *número Pi*.

Gráfico 50

Ejecución de mclapply en terminal CentOS



Fuente: El Autor

Se muestra en la siguiente Tabla 10 un resumen de cada una de las variables y funciones del script del *número Pi* con los parámetros de tipo, longitud, tamaño y valor.

Tabla 10

Resumen de Variables y Funciones

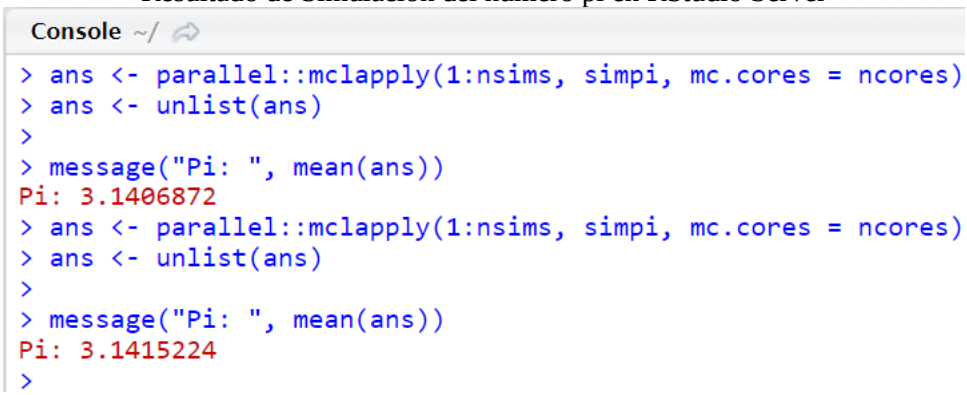
Nombre	Tipo	Longitud	Tamaño	Valor
ans	Numérico	1000	7.9 KB	Num [1:1000]
n	Numérico	1	56 B	10000
ncores	Entero	1	56 B	4L
nsims	Numérico	1	56 B	1000

simpi	Función	1	30.2 KB	Función (i)

Fuente: El Autor

Por medio de la consola de RStudio Server se presenta la ejecución del cálculo aproximado del *número Pi* con uso de la librería mclapply que se observa en el siguiente Gráfico 51.

Gráfico 51
Resultado de Simulación del número pi en RStudio Server



Fuente: El Autor

Se señala en la siguiente Tabla 11 el resumen de treinta ejecuciones del script *número Pi* en el entorno de RStudio Server que servirá para el análisis de desempeño del grid TesisMaster en el clúster OpenHPC.

Tabla 11
Resumen de Ejecución de la Simulación del número Pi

Número de Ejecución	Número de Simulaciones	Resultado Aproximado
1	[1:1000]	3.1406872
2	[1:1000]	3.1415224
3	[1:1000]	3.142086
4	[1:1000]	3.141432
5	[1:1000]	3.1417396
…	…	…
30	[1:1000]	3.1429675

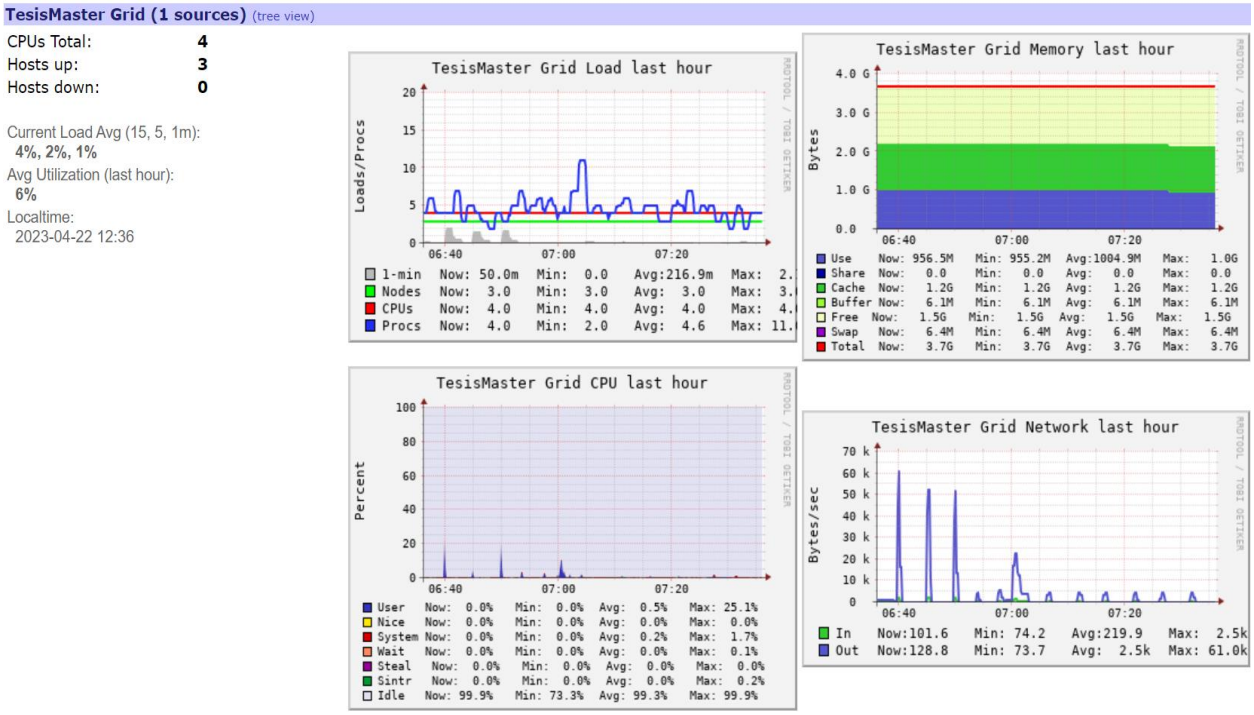
Fuente: El Autor

4.3.2. Análisis de Desempeño del Clúster

En este correspondiente análisis de desempeño se tomará como referencia la métrica de ***carga general*** del grid *TesisMaster* que obtiene una vision general del comportamiento de cada uno de los nodos que conforman el clúster OpenHPC.

Se analiza el comportamiento del grid con los parametros de consumo de memoria, carga de red y carga de procesamiento que se visualiza en el siguiente Gráfico 52 el consumo de recursos del grid en el clúster OpenHPC.

Gráfico 52
Análisis General de Carga del Grid TesisMaster

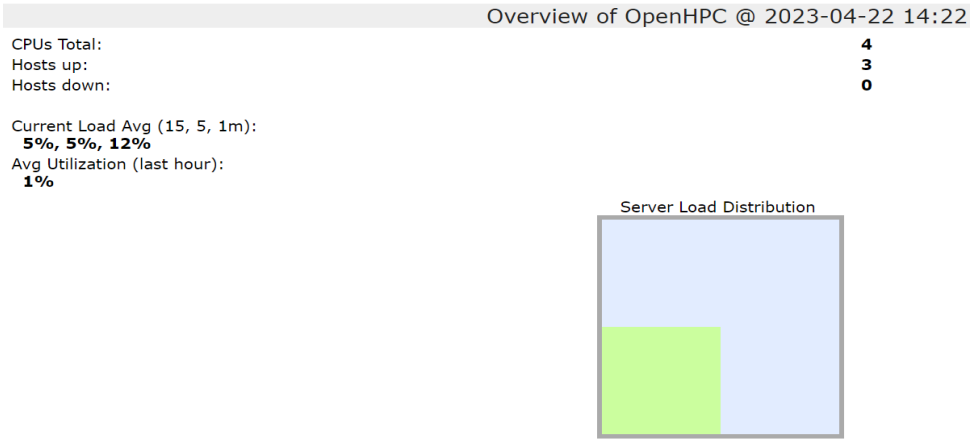


Fuente: El Autor

La distribución de carga del servidor refleja un promedio de carga del 24% que denota en *color fosforescente*, para una carga promedio actual del 12% en el reporte de actualización de última hora de uso que se divisa en el siguiente Gráfico 53.

Gráfico 53

Distribución de carga del servidor

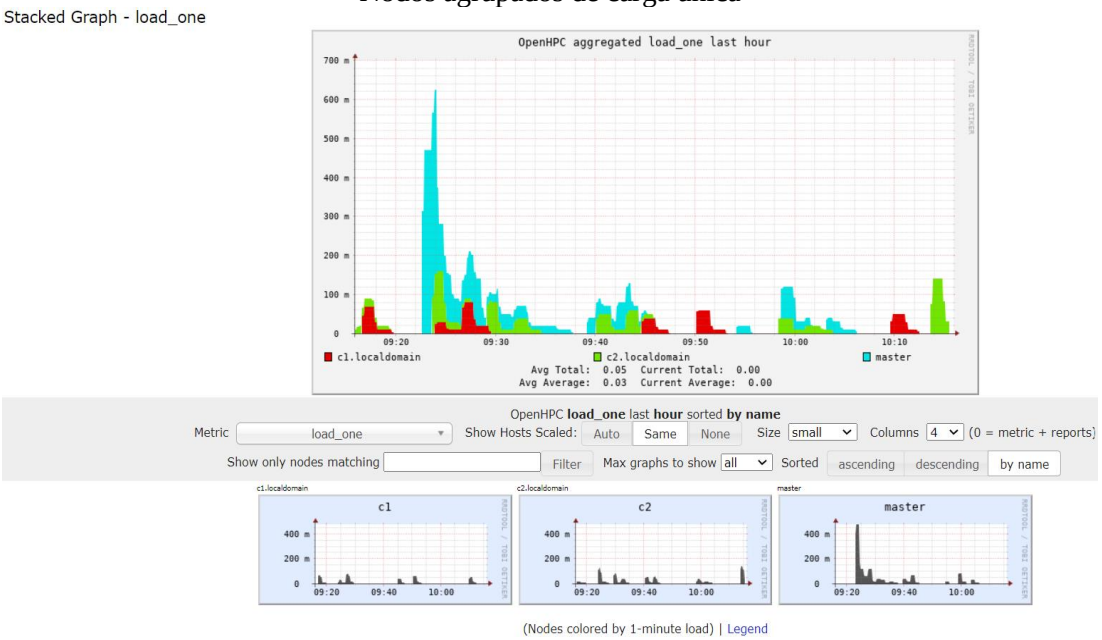


Fuente: El Autor

Se aprecia la carga de los nodos agrupados en el siguiente Gráfico 54 donde el nodo master proporciona una mayor carga debido a que distribuye la información a los demás nodos. También porque aloja la configuración del clúster OpenHPC y el entorno de RStudio Server.

Gráfico 54

Nodos agrupados de carga única

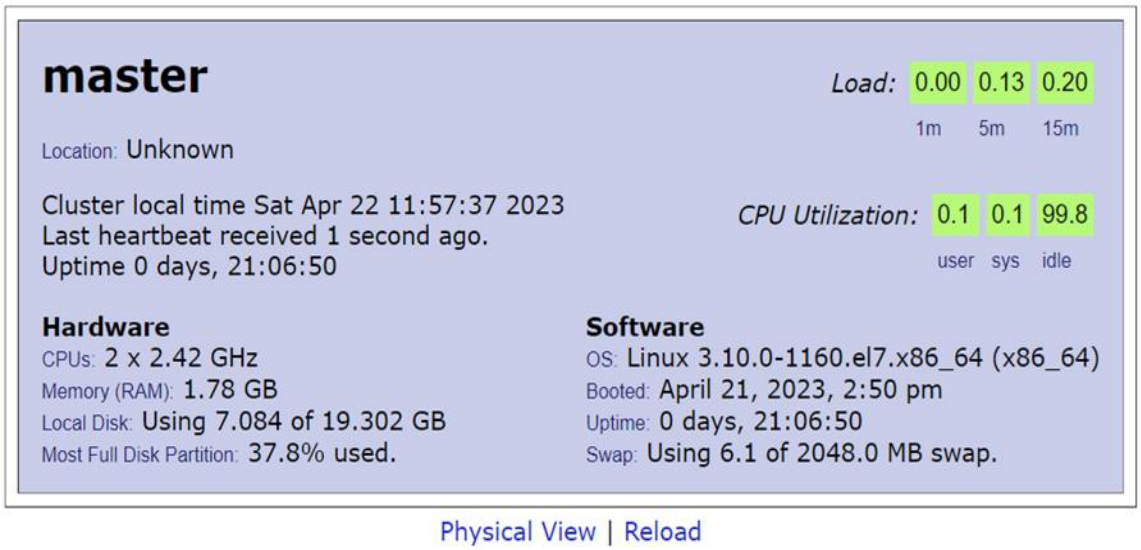


Fuente: El Autor

Se presenta en el siguiente Gráfico 55 un resumen general de las características del nodo master como el hardware, software, carga y uso del CPU. Se aprecia también el reporte referente a la

carga y uso de procesamiento que monitorea en tiempo real el desempeño del clúster con la finalidad de poder brindar una distribución de carga óptima a través del grid.

Gráfico 55
Resumen de Carga del Nodo Master



Fuente: El Autor

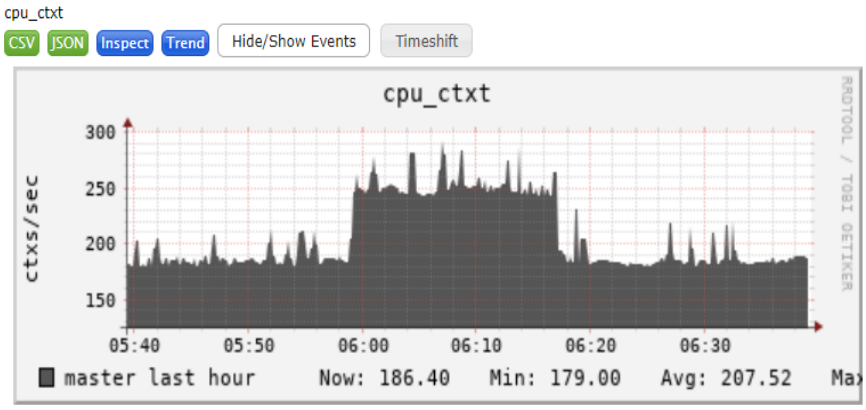
4.3.3. Análisis de métricas de desempeño de carga única del nodo master

Con los datos obtenidos de las métricas de procesamiento, disco, memoria y red se analiza el desempeño del nodo master para monitorear el comportamiento de carga que requiere una atención oportuna en el clúster. A continuación, se presenta cada una de las métricas del nodo master del grid *TesisMaster*.

Métricas de CPU.

La presente métrica refleja el contexto de carga que almacena o restaura el estado del CPU en el que se comparten los multiprocesos de manera única en el CPU del nodo master. En este caso la carga máxima de multiprocesos alcanza el intervalo promedio de 207.52 ctxts/seg que se muestra en el siguiente Gráfico 56.

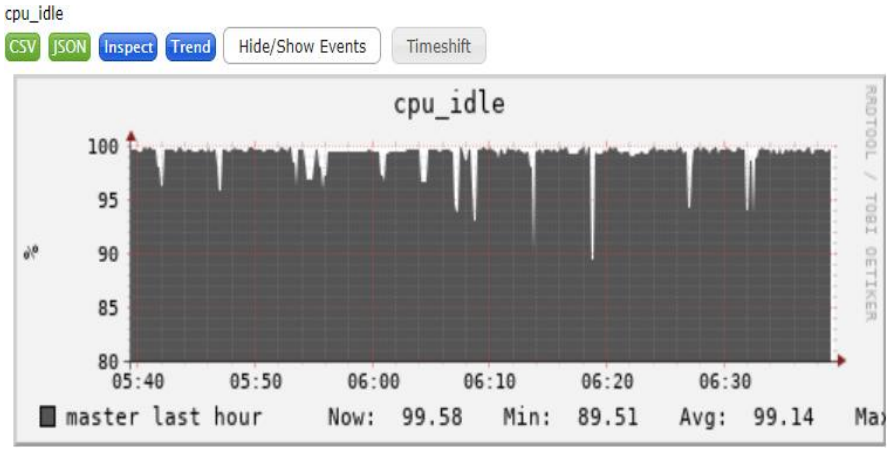
Gráfico 56
Contexto CPU



Fuente: El Autor

Se observa en el siguiente Gráfico 57 que hay un 99.14% de carga en baja prioridad para que no genere conflicto con los procesos que requieran ejecuciones en prioridad normal, lo que facilita un desempeño optimo sobre los otros procesos.

Gráfico 57
Inactividad de Procesos del CPU

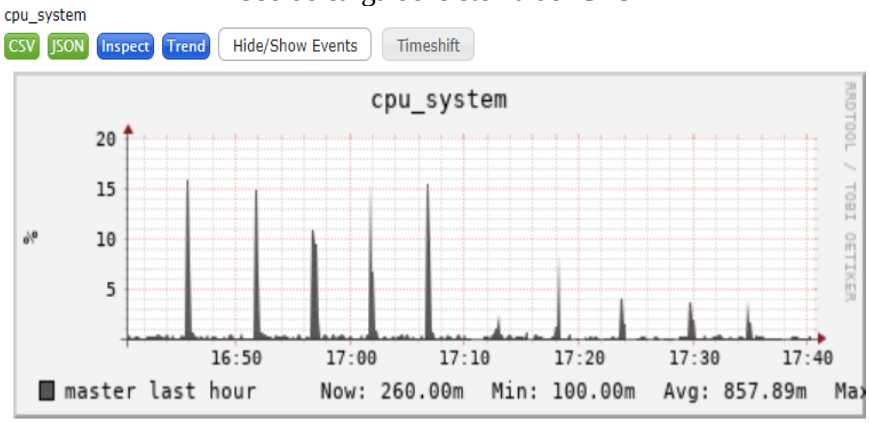


Fuente: El Autor

Se analiza en el siguiente Gráfico 58 que el uso de carga de 857.89m refleja un comportamiento de ejecución de procesos del 16%, lo que favorece un desempeño fluído en la carga del sistema.

Gráfico 58

Uso de carga del sistema del CPU

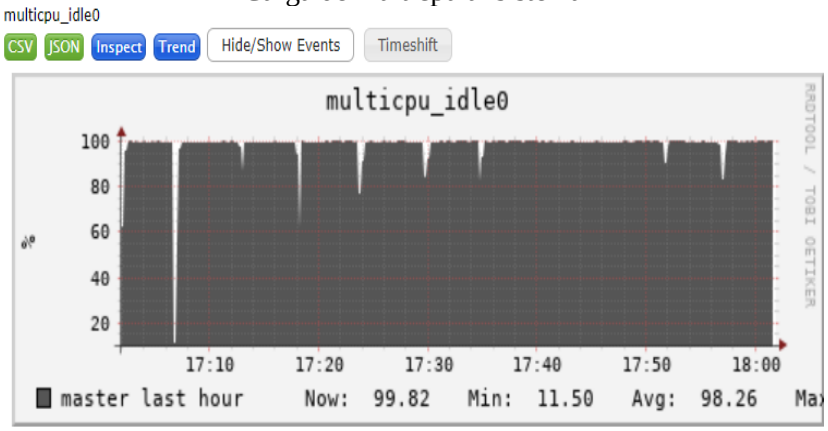


Fuente: El Autor

Se divisa en el siguiente Gráfico 59 una ocurrencia de carga del 98.26% de multiprocesos en prioridad baja para dar prioridad a los multiprocesos que requieran prioridad de ejecución alta.

Gráfico 59

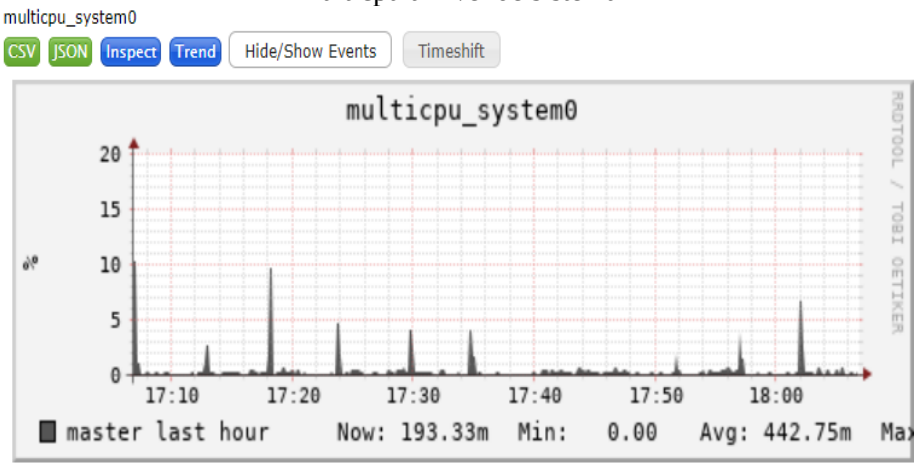
Carga de multicpu al sistema



Fuente: El Autor

Se refleja en el siguiente Gráfico 60 una carga de multiprocesos de 442.75m con un porcentaje de uso del 11% en las ejecuciones a nivel de sistema.

Gráfico 60
Multicpu a nivel de sistema

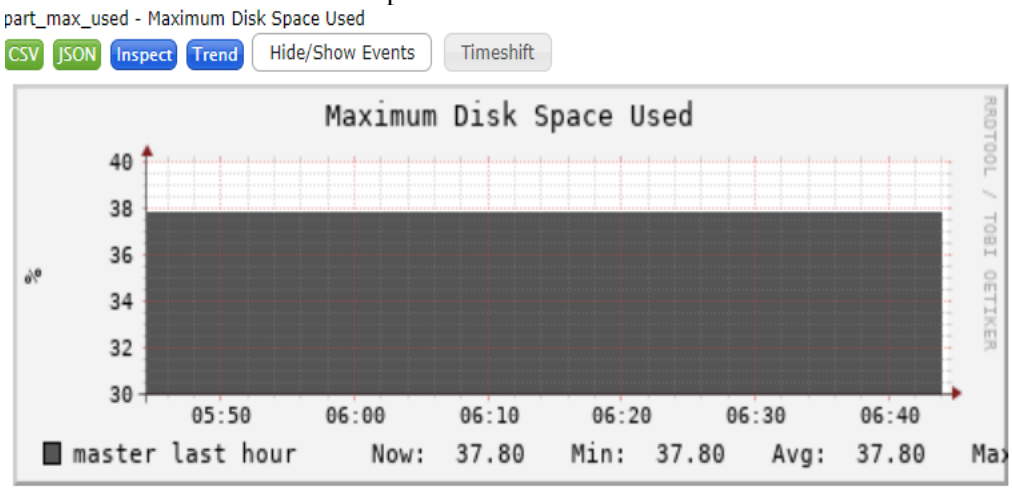


Fuente: El Autor

Métricas de Disco

En esta métrica se analiza que el máximo usado de almacenamiento en el host master es del 37.8% como se puede visualizar en el siguiente Gráfico 61.

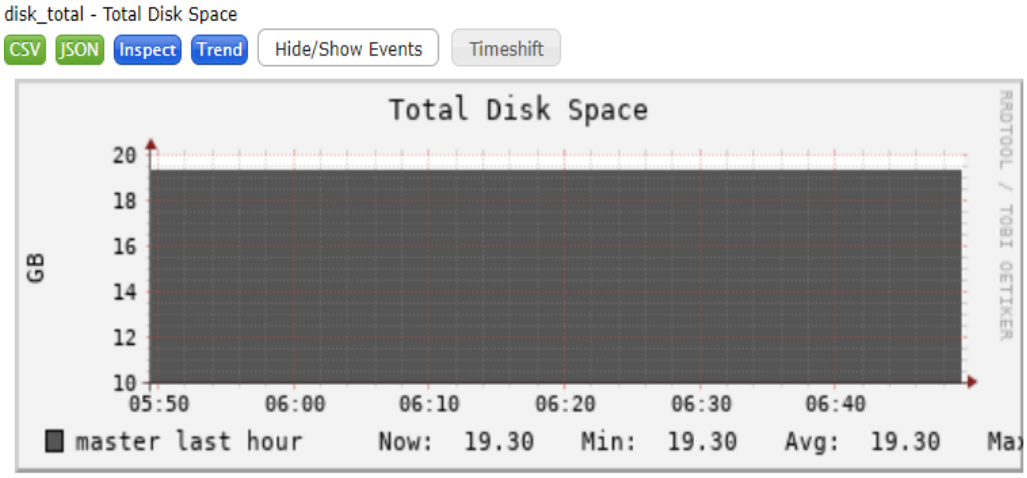
Gráfico 61
Espacio usado en el disco



Fuente: El Autor

Se observa en el siguiente Gráfico 62 que el total de espacio disponible del disco que es de 19 GB en el nodo master.

Gráfico 62
Espacio Total del disco

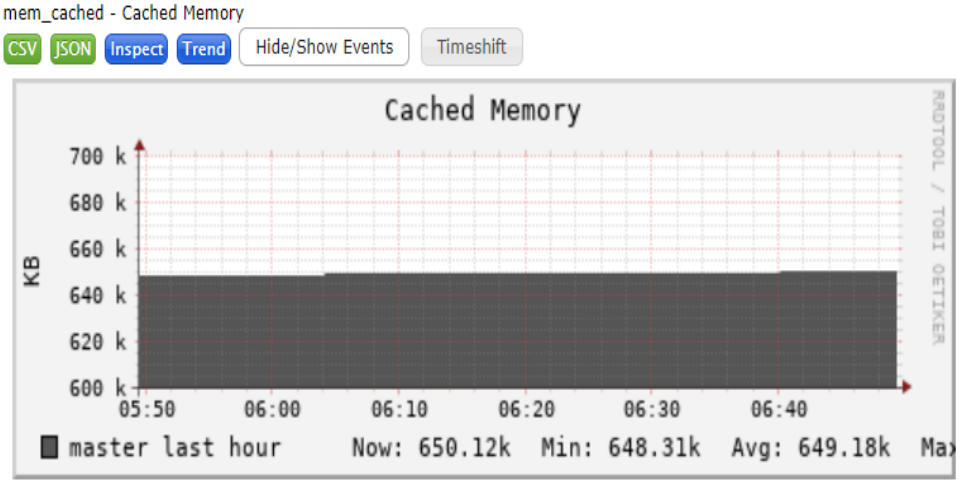


Fuente: El Autor

Métricas de Memoria

Se analiza que la carga de memoria cache almacenada de datos es de promedio 649.18 KB como se puede visualizar en el siguiente Gráfico 63.

Gráfico 63
Memoria Cache

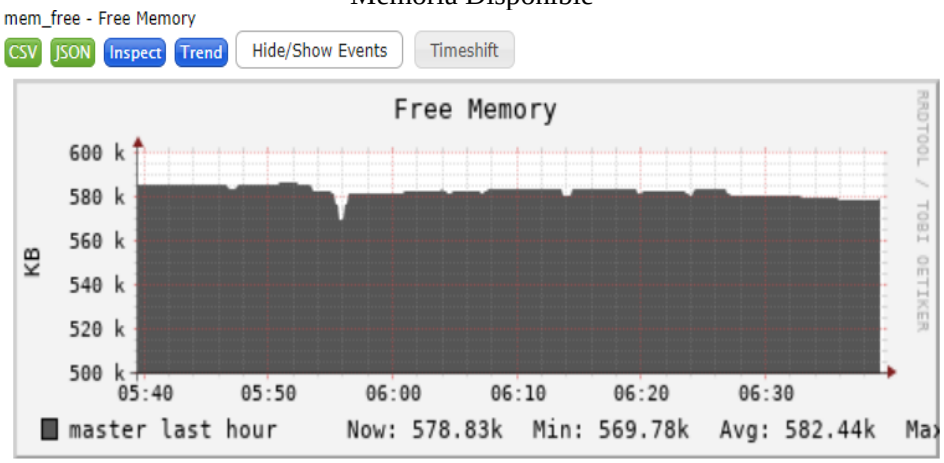


Fuente: El Autor

Se examina en el siguiente Gráfico 64 que la cantidad de memoria ram disponible es de 582.44 K, esto favorece el aprovechamiento al máximo del desempeño del grid.

Gráfico 64

Memoria Disponible



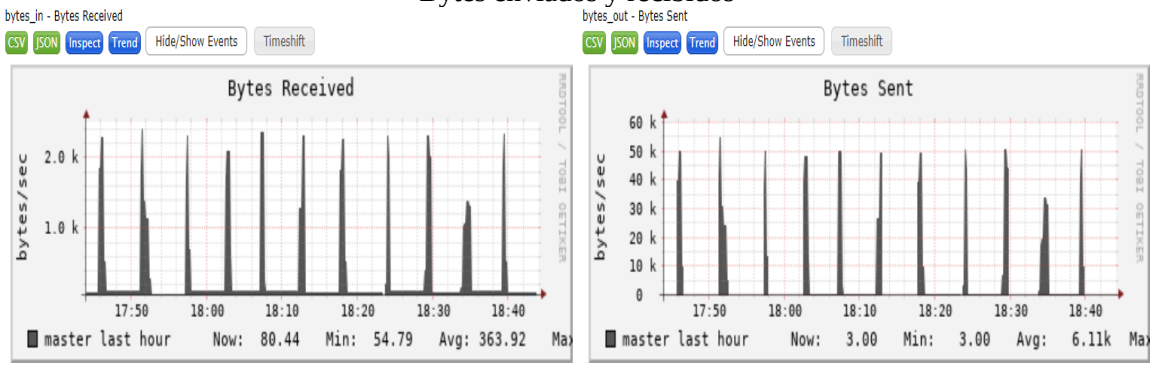
Fuente: El Autor

Métricas de Red

Se aprecia en el siguiente Gráfico 65 que el tráfico de carga en bytes enviados y recibidos en el nodo master son alrededor 55 KB/seg y 2.3 KB/seg respectivamente. Lo que muestra una latencia de tráfico estándar en el grid para los bytes que se envían y reciben.

Gráfico 65

Bytes enviados y recibidos

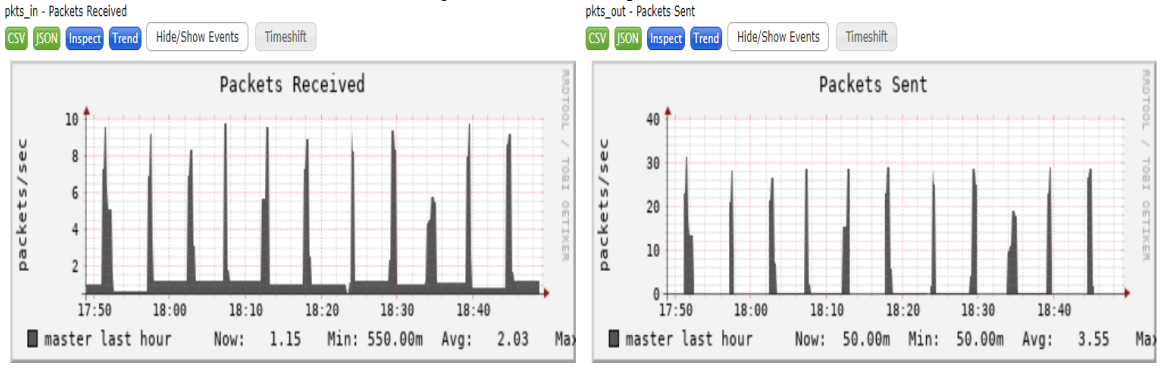


Fuente: El Autor

Se analiza que el trafico de paquetes es de 3.55 paquetes enviados y 2.03 paquetes recibidos respectivamente, lo que muestra en el Gráfico 66 un comportamiento equilibrado de envio y recibido paquetes.

Gráfico 66

Paquetes enviados y recibidos

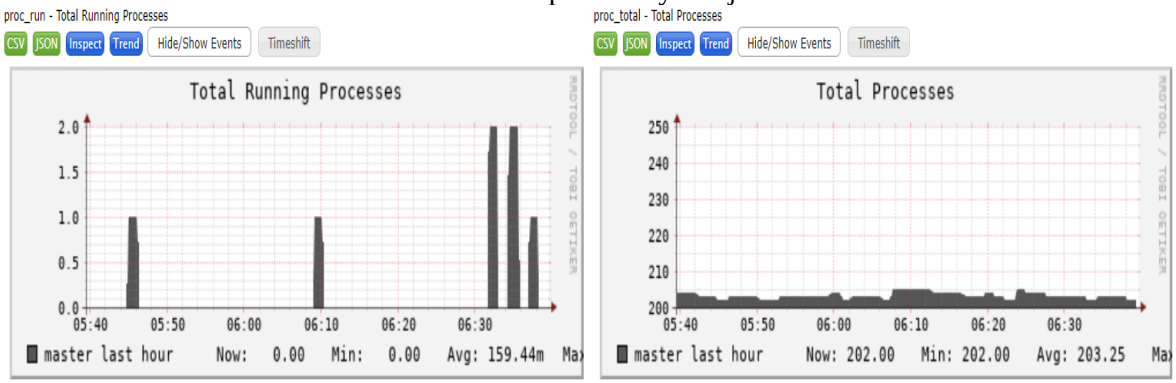


Fuente: El Autor

Se observa en el siguiente Gráfico 67 que la cantidad de procesos ejecutándose es 159.44m de un total de 203.25m, lo que denota un desempeño eficiente en el comportamiento del grid.

Gráfico 67

Total de procesos y de ejecución



Fuente: El Autor

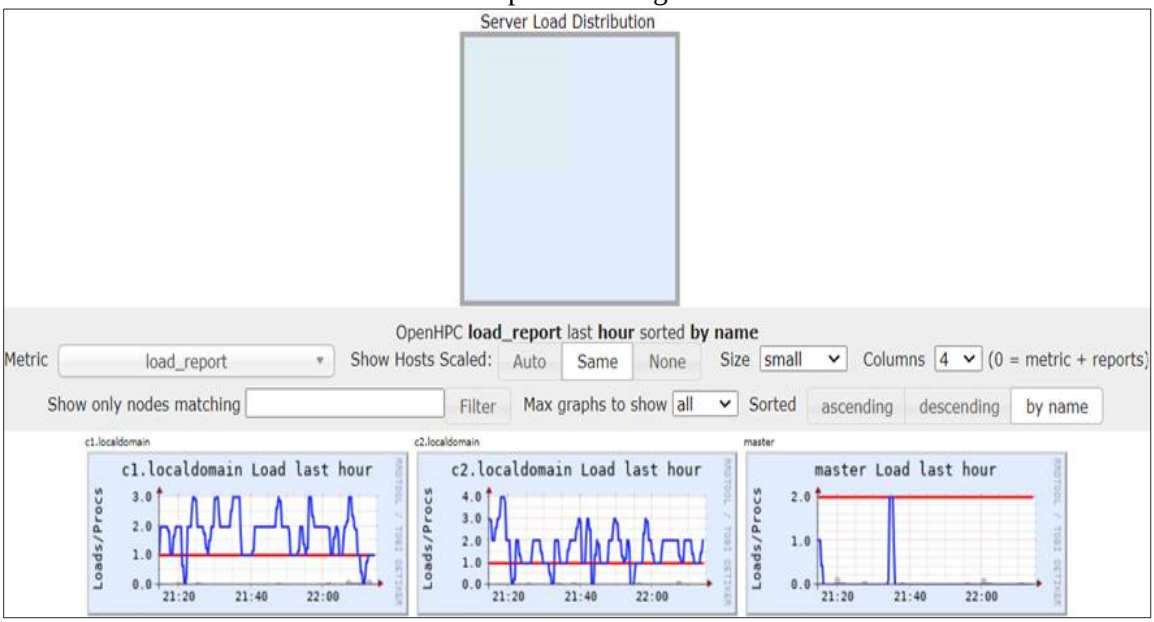
CAPÍTULO V: ANÁLISIS DE RESULTADOS

En el presente apartado se procede con el análisis de los resultados que arrojó las métricas de reportes del grid *TesisMaster*. Los datos de la muestra de estudio de la plataforma HPC estuvo conformada por cada uno de los nodos que conforman el grid construido con la herramienta OpenHPC y gestionada por el entorno de Ganglia.

5.1. Reporte de Carga

El reporte de carga en el Gráfico 68 refleja un ligero incremento en las cargas de trabajo en el nodo C2 con referencia al nodo C1. Donde ambos nodos C1 y C2 comparten la misma cantidad de procesos para un desempeño constante gestionados por el nodo master que realiza a manera de ráfaga. Lo que implica una sobrecarga perpendicular leve en los nodos C1 y C2 respectivamente.

Gráfico 68
Reporte de Carga



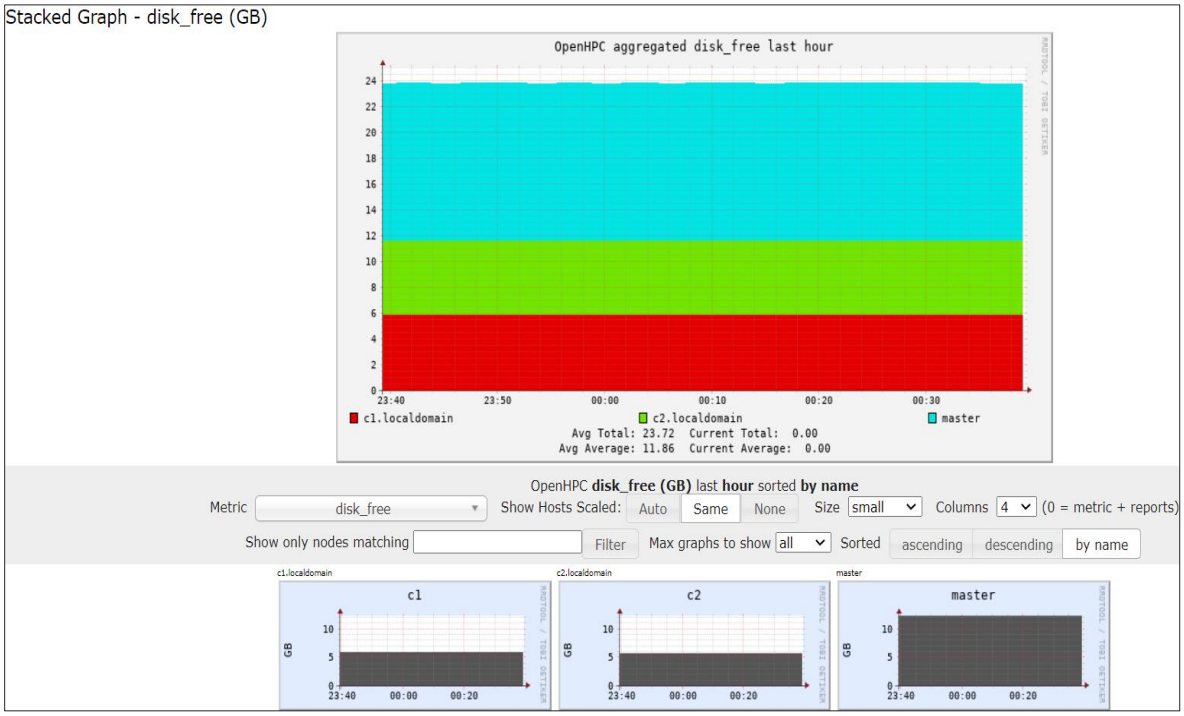
Fuente: El Autor

5.2. Espacio libre en Almacenamiento

Se visualiza en el siguiente Gráfico 69 un total de almacenamiento de 36 GB en todo el grid luego de realizadas las respectivas configuraciones de instalación, se deduce un total disponible de 23.72

GB con un promedio libre de 11.86 GB. El nodo master requiere mayor disponibilidad de almacenamiento por ser el que realiza las ejecuciones del script de aproximación del *número Pi* en el entorno RStudio Server y donde se aloja la plataforma de Ganglia.

Gráfico 69
Almacenamiento libre

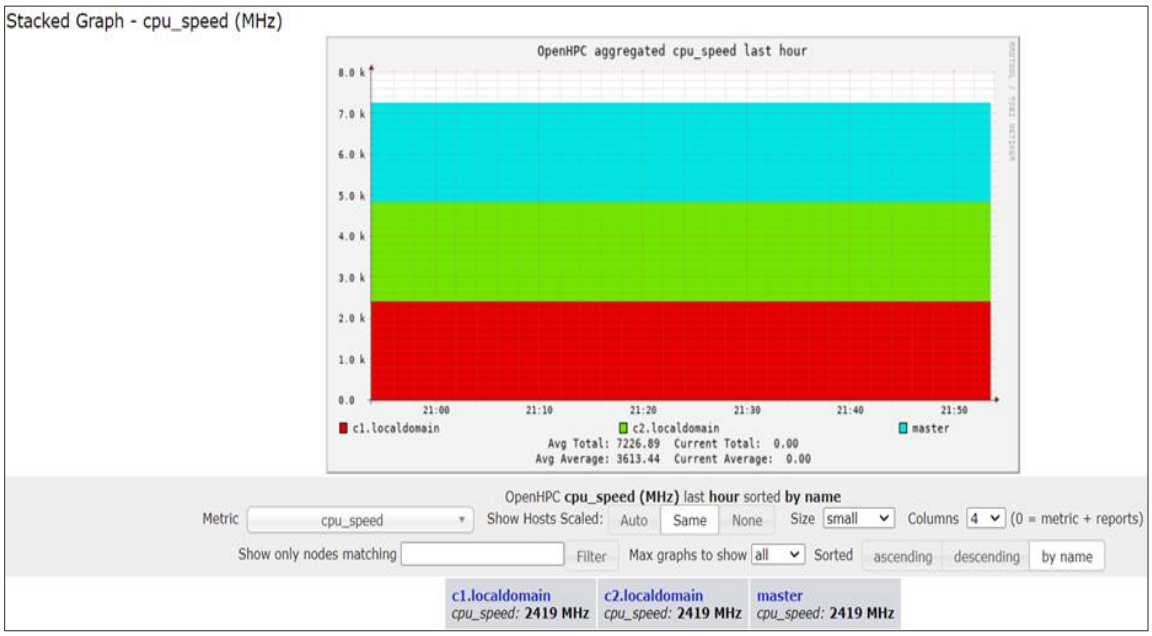


Fuente: El Autor

5.3. Velocidad de Procesamiento

Se refleja un total de velocidad de procesamiento de 7226.89 MHz para todo el grid, donde cada uno de los nodos posee una velocidad estándar de 2419 MHz, lo que facilita el aprovechamiento al máximo de los recursos del host master. Se aprovecha al máximo los recursos porque tiende un comportamiento de aumento en el uso del procesamiento en las cargas de trabajo como se observa en el siguiente Gráfico 70.

Gráfico 70
Velocidad del CPU

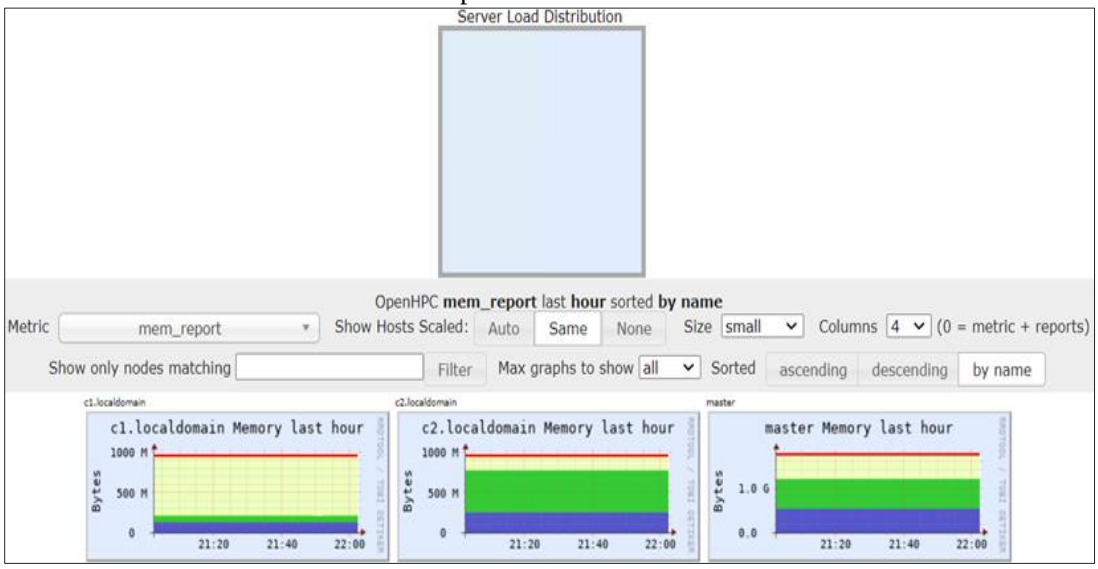


Fuente: El Autor

5.4. Reporte de Memoria

Se muestra en el Gráfico 71 un comportamiento de memoria compartida equitativa en el nodo master para facilitar el almacenamiento de información en la ejecución de procesos, lo que representa un ligero aumento en su memoria cache considerable en el nodo C2. Refleja un compartimiento de información ligeramente reducido al nodo master porque recibe mayor cantidad de información. En contraste, el nodo C1 tiende una mayor disponibilidad de memoria libre y uso compartido para una carga de trabajo reducida en referencia al nodo C2.

Gráfico 71
Reporte de Memoria

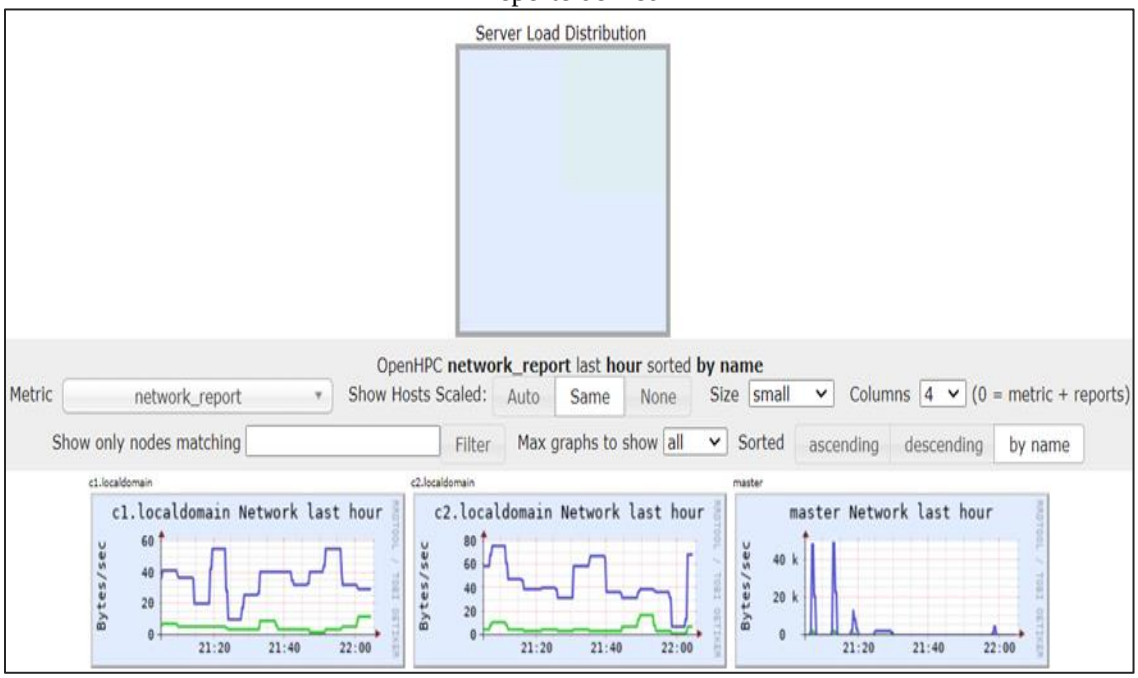


Fuente: El Autor

5.5. Reporte de Red

La red del grid refleja un comportamiento de transferencia de bytes constante entre los nodos C1 y C2 que muestra una mayor tasa de transferencia de 60 Kb/s en el nodo C2 con respecto al nodo C1 que realiza una transferencia de 40 Kb/s. Mientras en el nodo master se visualiza un envío de datos descendente, pero con un ligero incremento conforme avance el envío de bytes que se observa en el siguiente Gráfico 72.

Gráfico 72
Reporte de Red



Fuente: El Autor

5.6. Reporte de Paquetes

La transferencia de bytes reflejado en el Gráfico 72 se compara de manera perpendicular con la recepción de paquetes. Arroja una ligera variación descendente de paquetes en el nodo C1 con respecto al nodo C2, pero con una relevante recuperación en el envío de paquetes. El nodo master tiende a un envío de paquetes de manera reducida hacia los nodos C1 y C2 que se puede divisar en el Gráfico 73.

Gráfico 73
Reporte de Paquetes

Fuente: El Autor

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- La HPC motiva a implementar centros de alto rendimiento en las distintas organizaciones a nivel mundial porque proporciona una ventaja competitiva en el área de investigación científica. Realiza velozmente mejoras en la ejecución de tareas debido a su procesamiento paralelo. La implementación de un centro HPC implica un grado de inversión alta en recursos en el plano económico y profesional, lo que compromete un aprendizaje constante en el manejo de un gran volumen de datos que las organizaciones deben considerar.
- Con la información recabada se concluye que las plataformas HPC poseen servidores de cómputo conectados en red que se llaman nodos, sus requerimientos de computación, red y almacenamiento proveen datos hacia y desde los servidores. Su alto rendimiento influye en la ejecución de tareas en computación paralela desde servidor hacia los nodos que conforman el clúster. Genera un beneficio significativo en la reducción de tiempo en la ejecución de cargas de trabajo, prontitud en los cálculos matemáticos que deben realizarse para optimizar los recursos con la finalidad de aplicarlo en el campo manufacturero, gubernamental, medico, empresarial, etc.
- Mediante la tecnología de virtualización se consiguió aprovechar la ejecución de varias máquinas virtuales en un solo procesador. Con la herramienta OpenHPC que provee una serie de componentes se pudo construir un clúster que facilitó el acceso para gestionar herramientas de aprovisionamiento y monitoreo la plataforma HPC. Con los componentes facilitados por OpenHPC se logró hacer uso de la aplicación Ganglia que sirvió para el análisis de desempeño del grid TesisMaster.
- A través de la aplicación Ganglia Monitoring se consiguió obtener los resultados para

realizar el análisis de desempeño de la plataforma HPC y supervisar el comportamiento del grid. Lo que permitió verificar las estadísticas presentadas en los parámetros de almacenamiento, red y procesamiento. Con Ganglia también se hizo un análisis minucioso sobre un resumen de métricas proporcionadas por la aplicación porque favorece a las organizaciones que poseen sistemas HPC para monitorear y evaluar el desempeño de sus clústeres.

- Con la ejecución del script de *número Pi* se logró la simulación de aproximación del punto flotante en el desempeño de la plataforma HPC. El entorno RStudio server facilitó su verificación de aproximación mediante su librería paralela mclapply para proceder con el análisis de desempeño del grid en la plataforma de Ganglia.

Recomendaciones

- Con el avance tecnológico de información que se incrementa de manera exponencial, se recomienda a las organizaciones de investigación en Latinoamérica y Ecuador implementar sistemas de HPC para fomentar el avance tecnológico en el ámbito investigativo y científico a fin de brindar soluciones óptimas en favor de la humanidad en general.
- Es recomendable considerar que las implementaciones en sistemas HPC conlleva una inversión económica muy significativa de recursos. Pero a su vez, las organizaciones que logren su implementación, generarán una ventaja competitiva en el campo científico y tecnológico que se podrá reflejar en las soluciones HPC que brinden a la comunidad.
- Se recomienda optar por la informática HPC en RHEL debido a que brinda una plataforma confiable y eficiente al momento de acceder a las diversas herramientas que ofrece OpenHPC porque implica un ahorro de coste considerable al momento de construir un clúster en una herramienta opensource.
- Es recomendable recurrir a la aplicación de Ganglia Monitoring porque permite supervisar y evaluar de manera oportuna el desempeño del grid, con la finalidad de realizar un análisis sobre que nodo se encuentra disponible o está haciendo uso de recursos de cpu y memoria. Con los datos proporcionados por Ganglia se puede obtener un estado de comportamiento del grid en el clúster de OpenHPC.
- Se recomienda el uso de RStudio Server porque ofrece una interfaz amigable para el usuario y poder realizar simulaciones de algoritmos basados en HPC. De manera adicional se puede optar por la simulación en la consola de comandos de CentOS para verificar la aproximación del *número Pi*.

TRABAJOS A FUTURO

En un entorno físico óptimo donde se instale un centro HPC se propondría plantear un modelo de servidores de almacenamiento para cargas de trabajo que pueda ser usado para futuras consultas o referencias para trabajos en HPC de la PUCE.

Mediante la proyección de la HPC se la podría implementar en un centro HPC de la PUCE con las debidas planificaciones que conlleva en la inversión de recursos económicos y tecnológicos para su puesta en marcha que se requiera. Se deberá adquirir diversos equipos de hardware y software que se ofertan en el mercado tales como, servidores HPC, chasis de rack, equipos de interconexión de redes, procesadores y aceleradores gráficos RAM y equipos de almacenamiento.

Los proyectos en HPC podrían incorporar herramientas de monitoreo como ganglia u otras herramientas disponibles en el mercado para evaluar minuciosamente en el comportamiento de los clústeres en tiempo real. Con esta herramienta se alcanzaría a realizar un análisis para la recolección de datos sobre el comportamiento de los grid que se vayan a implementar a futuro. Al ser una plataforma HPC escalable se podrían añadir más nodos conforme siga expandiéndose el desempeño del grid.

Con los conocimientos adquiridos sobre servidores HPC los desarrolladores de proyectos de software podrían realizar pruebas y administrar los cálculos científicos de alto rendimiento. Se podría implementar clústeres de servidores HPC para generar aportes significativos a la comunidad científica en los diversos campos de aplicación que requiera su uso.

REFERENCIAS

Alvear, J. O. (13 de 07 de 2018). *HPC con R para Investigadores*. https://bookdown.org/content/98faea5e-842f-47ba-be83-493572e28e7b/paralelizacion-con-r-en-el-cluster-parallel.html

Andrés, B. A. (2021). Solving Scheduling Problems with Ant Colony Optimization Metaheuristic and High Performance Computing. *Solving scheduling problems with ant colony optimization metaheuristic and high performance computing.* Universidad YACHAY TECH , Urcuqui , Ecuador . https://doi.org/http://repositorio.yachaytech.edu.ec/handle/123456789/375

CARLA. (2023). *La Conferencia Latinoamericana de Computación de Alto Rendimiento llega a Colombia*. https://www.carla2023.org/

Corral García, J. (2021). *Paralelización automática y estrategias de desarrollo del código eficiente para aumentar el rendimiento en centros de supercomputación [Tesis Doctoral, Universidad de Extremadura].* Repositorio Institucional. http://hdl.handle.net/10662/12693

DataCenterMarket. (5 de 1 de 2022). *Conoce el laboratorio de innovación de HPC e IA de Dell Technologies*. https://www.datacentermarket.es/mercado/hpc/1130512032609/conoce-laboratorio-de-innovacion-de-hpc-ia-de-dell-technologies.1.html

Fabián Bernal, C. A. (2020). *Programación Paralela*. http://ferestrepoca.github.io/paradigmas-de-programacion/paralela/paralela_teoria/

Fabián Bernal, C. A. (2021). *Programación Paralela*. http://ferestrepoca.github.io/paradigmas-de-programacion/paralela/paralela_teoria/

IBM. (2023). *¿Qué es la computación de alto rendimiento (HPC)?* https://www.ibm.com/es-es/topics/hpc

InsideHPC. (2023). *Archives for Industry Segments*. https://insidehpc.com/category/industry-segments/

James A. Ang, D. J. (2022). New Horizons for High-Performance Computing. *The IEEE Computer Society*, 7. https://doi.org/10.1109/MC.2022.3200859

NLHPC. (2023). *Estado del cluster*. https://dashboard.nlhpc.cl/

NVIDIA DEVELOPER. (2023). *Ganglia Monitoring System*. https://developer.nvidia.cn/ganglia-monitoring-system

openHPC. (17 de 07 de 2020). *Community building*. https://openhpc.community/

Oracle . (2023). *¿Qué es la computación de alto rendimiento, o HPC?* https://www.oracle.com/cl/cloud/hpc/what-is-hpc/

RCAB. (2023). *High Performance Computing (HPC)*. https://about.illinoisstate.edu/rcab/hpc/

RCAB. (2023). *High Performance Computing (HPC)*. https://about.illinoisstate.edu/rcab/

RedCLARA. (2023). *Historia de RedCLARA*. https://redclara.net/index.php/es/somos/redclara-la-organizacion/historia-de-redclara

Rivera, B. L. (2017). Análisis de rendimiento de un clúster HPC y, arquitecturas manycore y multicore. *Repositorio Instucional* . UNIVERSIDAD DE CUENCA, Cuenca, Azuay, Ecuador. https://doi.org/http://dspace.ucuenca.edu.ec/handle/123456789/28554

TOP 500. (2022). *INTRODUCTION AND OBJECTIVES*. https://www.top500.org/project/introduction/

Top500. (11 de 2022). *List of November 2022*. https://www.top500.org/lists/top500/2022/11/

USCbiostats. (18 de 02 de 2021). *USC Division of Biostatistics.*

https://github.com/USCbiostats/slurmr-workshop/blob/master/02-mclapply.R

Velazquez, J. A. (2017). *Construcción y diseño de un clúster tipo HPC virtualizado [Tesis de maestria, Instituto Politecnico Nacional].* Repositorio Institucional. http://tesis.ipn.mx/handle/123456789/23630

vmWARE. (2023). *Virtualización de informática de alto rendimiento (HPC).* https://www.vmware.com/es/solutions/high-performance-computing.html

Yon, G. G. (19 de 03 de 2021). *Workshop on HPC with Slurm, R, and the slurmR package.* USCbiostats: https://github.com/USCbiostats/slurmr-workshop

Yuankun Fu, Y. F. (2022). *Virtualizing High Performance Computing (HPC) on VMware vSphere® 7 Using Intel® Select Solution for HPC*. https://core.vmware.com/resource/virtualizing-high-performance-computing-hpc-vmware-vsphere%C2%AE-7-using-intel%C2%AE-select-solution#overview

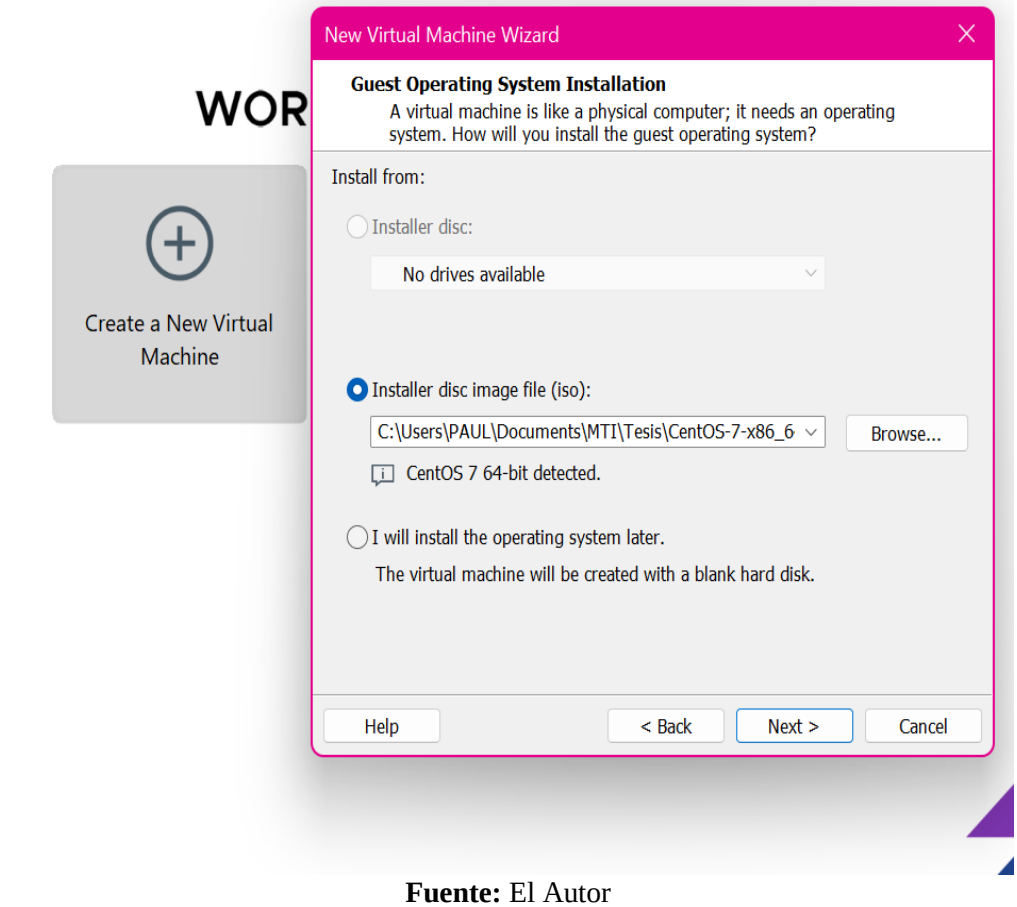
ANEXOS

ANEXO 1. Instalación del SO CentOS 7 en vmWARE Workstation Pro 17

En este apartado de ANEXO 1 se procede con los siguientes pasos principales para la instalación del SO CentOS 7 en los nodos TesisMaster, C1 y C2 en las máquinas virtuales.

En la página principal del vmWARE se da click en el botón de ***"Create a New Virtual Machine"***. Luego se busca la imagen iso del correspondiente SO que en este caso es CentOS 7 versión x86_64 Minimal y click en ***"Next"*** para continuar con la instalación como se observa en el siguiente Gráfico 74.

Gráfico 74
Creación de Nodos Virtuales

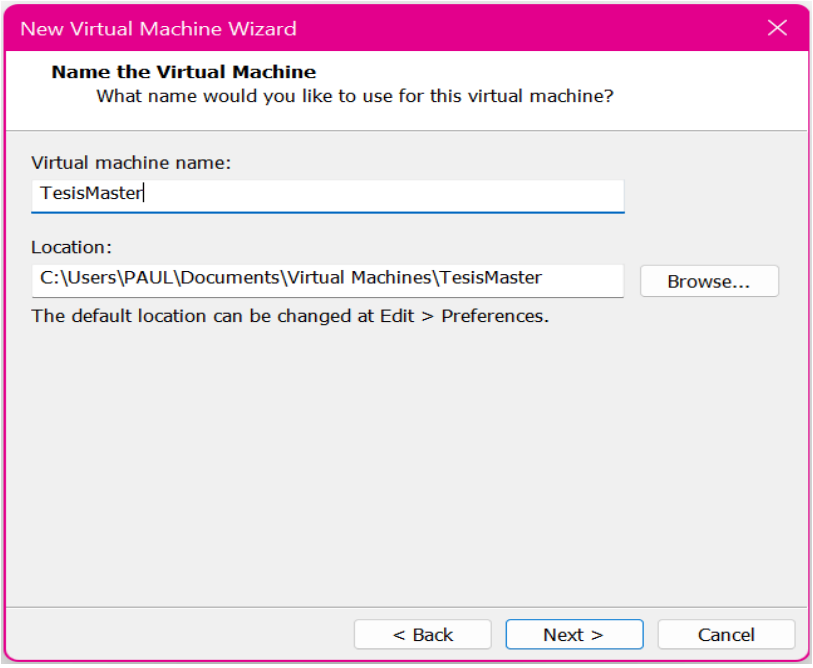


Fuente: El Autor

Luego se da nombre a la máquina virtual, como referencia se toma la instalación para el nodo ***TesisMaster***, posterior se da click en el botón ***"Next"*** para continuar con la instalación, como se

puede apreciar en el siguiente Gráfico 75.

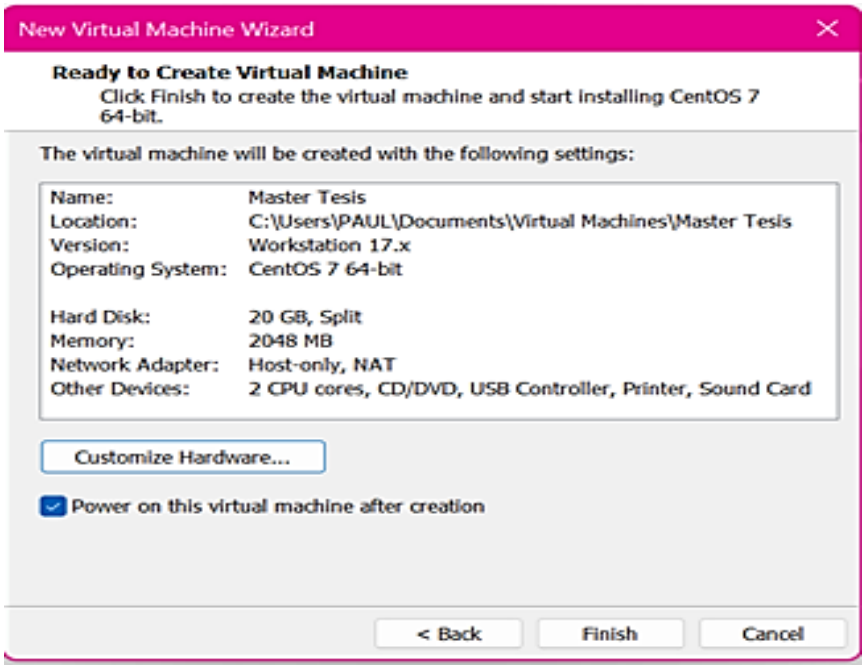
Gráfico 75
Nombre del Nodo Virtual



Fuente: El Autor

Una vez digitado el nombre de la máquina virtual, se puede visualizar las configuraciones de la misma para luego dar click en ***"Finish"*** y proceder con la instalación del SO como se puede visualizar en el siguiente Gráfico 76.

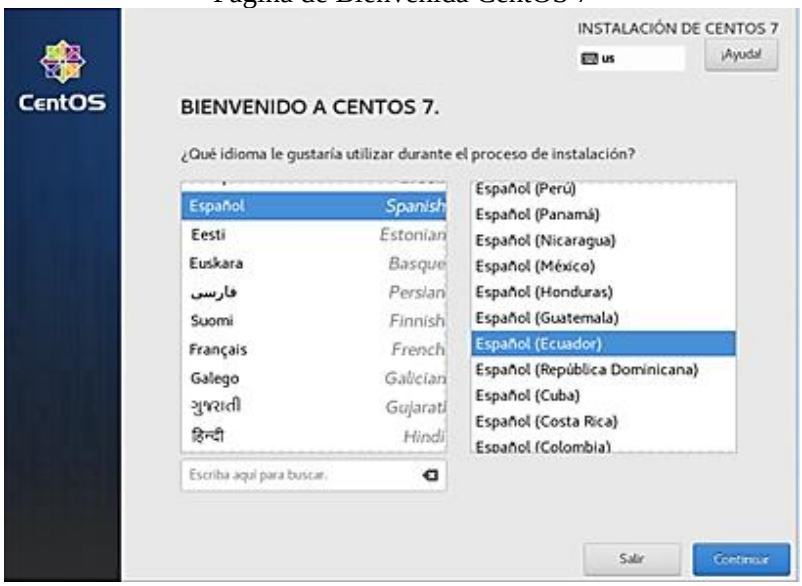
Gráfico 76
Información del Nuevo Nodo Virtual



Fuente: El Autor

Ya iniciado el proceso de instalación se realiza la elección del respectivo idioma que en este caso es el ***español*** y se da click en el botón ***"Continuar"*** como se puede divisar en el siguiente Gráfico 77.

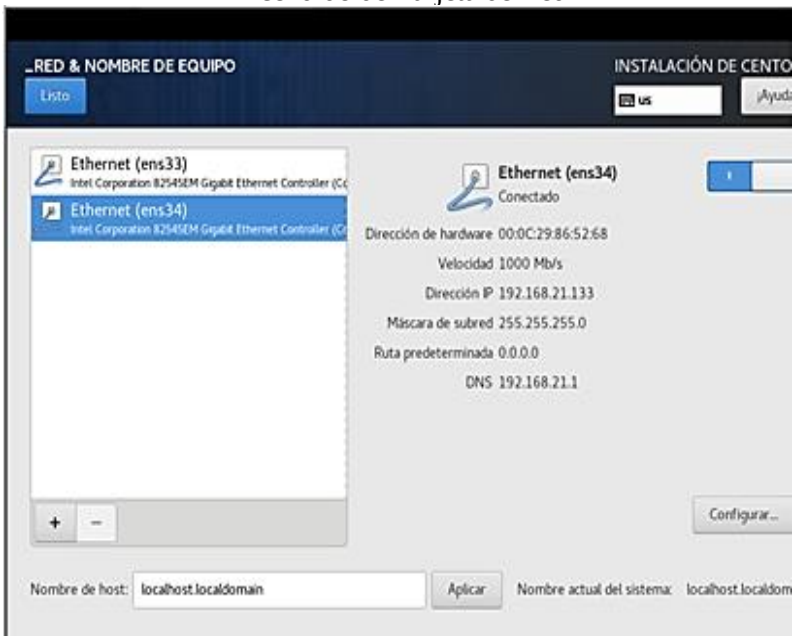
Gráfico 77
Página de Bienvenida CentOS 7



Fuente: El Autor

En la parte correspondiente a ***"Resumen de la Instalación"*** se puede elegir las opciones de configuración de ***"Red & Nombre del Equipo"*** para proseguir al encendido de la tarjeta de la red como se puede visualizar en el siguiente Gráfico 78.

Gráfico 78
Encendido de Tarjeta de Red



Fuente: El Autor

Luego se verifica que cada uno de los parámetros de ***"Fecha y Hora"***, ***"Teclado"*** y ***"Destino de Instalación"*** estén correctamente seleccionados, posterior se da click en el botón ***"Empezar Instalación"*** como se visualiza en las siguientes imágenes como se puede visualizar en el siguiente Gráfico 79.

Gráfico 79
Resumen Instalación CentOS 7

Fuente: El Autor

Para finalizar la instalación del SO CentOS 7 se digita las credenciales de ***"Creación de Usuario"*** y ***"Contraseña de ROOT"*** para el inicio de sesión que finaliza con la instalación de todos los componentes del SO como se puede divisar en el siguiente Gráfico 80.

Gráfico 80
Configuración de CentOS 7

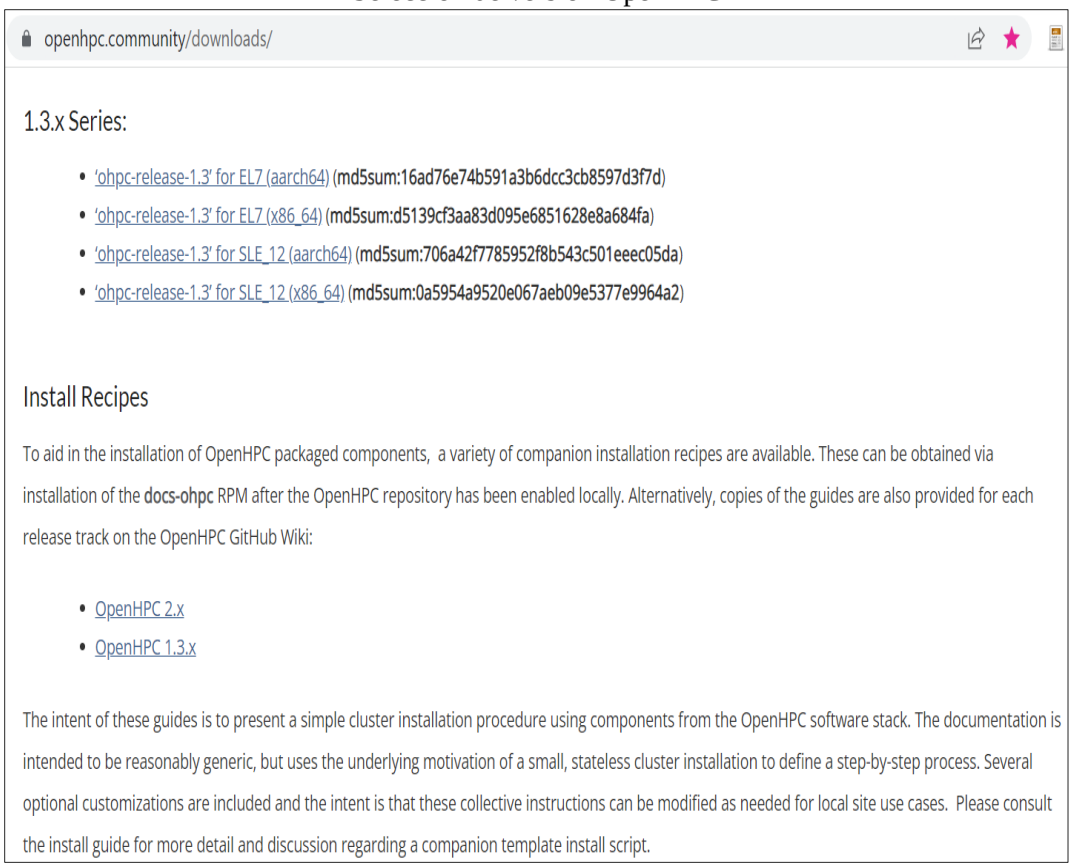
Fuente: El Autor

El mismo procedimiento correspondiente se realiza para los nodos cómputos de ***C1 y C2,*** la respectiva configuración del clúster se detalla en el siguiente apartado de ANEXO 2.

ANEXO 2. Instalación y Configuración de OpenHPC

Para este apartado del ANEXO 2 se realiza la instalación y configuración del cluster con el sistema OpenHPC desarrollado en distribución Red Hat Enterprise Linux que se encuentra disponible en la dirección https://openhpc.community/downloads/, para el desarrollo del presente proyecto de titulación se trabaja con la versión Open HPC 1.3.9, se da click en la parte de ***"OpenHPC 1.3x"*** para acceder al manual de instalación como se aprecia en el siguiente Gráfico 81.

Gráfico 81
Selección de versión OpenHPC



Fuente: El Autor

De referencia para la siguiente guía de instalación que se encuentra alojada en el recipiente de GitHub https://github.com/openhpc/ohpc/wiki/1.3.X se seleccionó para la versión del SO CentOS 7.7 el archivo en pdf de ***"Install Guide (with Warewulf + Slurm)"*** como se puede observar en el siguiente Gráfico 82.

Gráfico 82
Selección de Versión de Arquitectura de SO

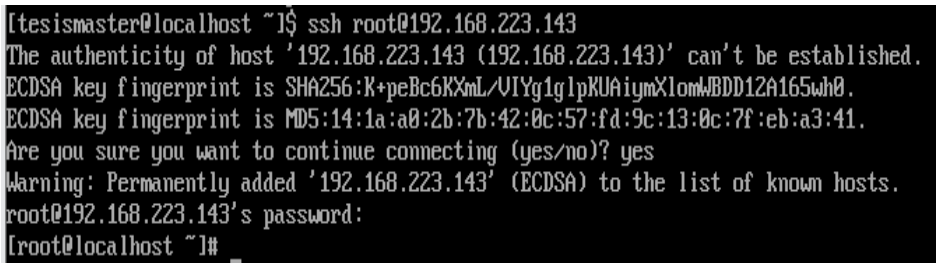
Fuente: El Autor

La presente guía muestra el procedimiento de instalación de un cluster simple con los componentes sistema Open HPC. Esto representa una serie de comandos requeridos para gestionar el clúster OpenHPC Linux con herramientas y recursos que se detallan a continuación.

Mediante conexión de *SSH (Secure Shell)* que es un protocolo de seguridad para gestionar el control a través de internet por medio de acceso de autenticación, para la presente configuración se accede a la dirección ip 192.168.223.143 que tiene salida a internet por el NAT como se puede divisar en el siguiente Gráfico 83.

Se accede al ip publica por SSH para realizar la configuración
[tesismaster@localhost ~]# ssh root@ 192.168.223.143

Gráfico 83
Acceso SSH al Nodo Master



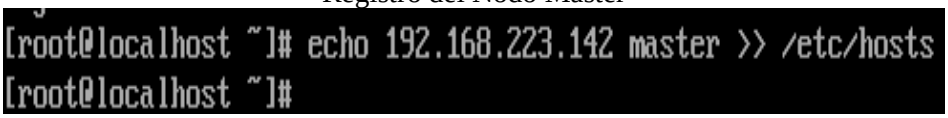
Fuente: El Autor

Para comenzar con la instalación de OpenHPC, primero se define la dirección ip interna del nodo

master 192.168.223.142 del host que en este caso es “master” en la dirección /etc/hosts con el siguiente comando como se muestra en el siguiente Gráfico 84.

Se asigna el nombre del host “master” a la red local
[root@localhost ~]# echo 192.168.223.142 master >> /etc/hosts

Gráfico 84
Registro del Nodo Master

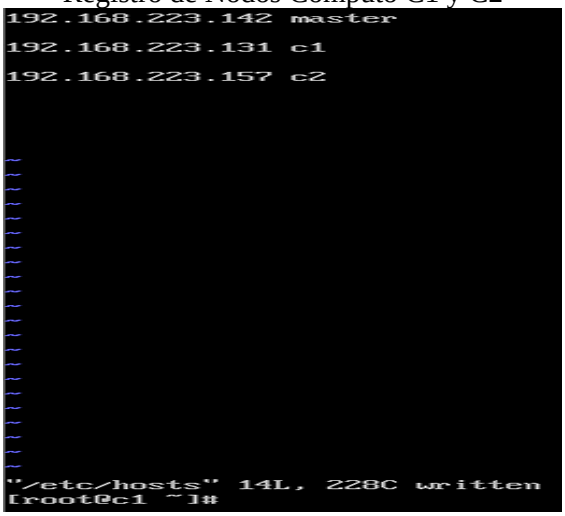


Fuente: El Autor

Otra forma de registrar los nodos es mediante el comando vi /etc/hosts y luego se procede con el registro de los nodos master, c1 y c2 con sus respectivas direcciones ip que se pueden apreciar en el siguiente Gráfico 85.

Se accede al directorio de host “master” para registrar los nodos
[root@localhost ~]# vi /etc/hosts

Gráfico 85
Registro de Nodos Computo C1 y C2



Fuente: El Autor

Se procede a *deshabilitar el firewall* porque puede ocasionar conflicto en la instalación de los componentes de OpenHPC mediante los siguientes comandos que se muestra en el siguiente Gráfico 86.

Se deshabilita el firewall del SO
[root@localhost ~]# systemctl disable firewalld
[root@localhost ~]# systemctl stop firewalld

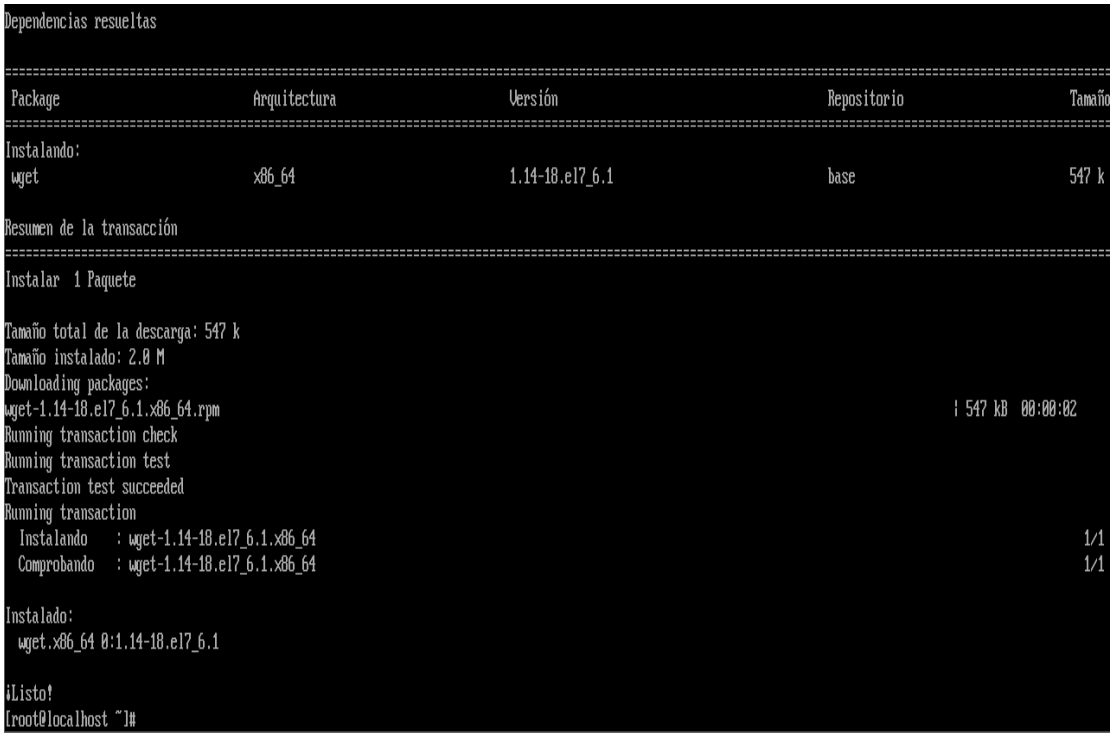
Gráfico 86
Deshabilitación del Firewall

Fuente: El Autor

Antes de instalar el repositorio se recomienda actualizar el SO y obtener nuevas actualizaciones mediante los siguientes comandos que se muestran en el siguiente Gráfico 87.

Se actualiza el SO
[root@localhost ~]# yum -y update
[root@localhost ~]# yum -y install wget

Gráfico 87
Actualización del SO CentOS 7

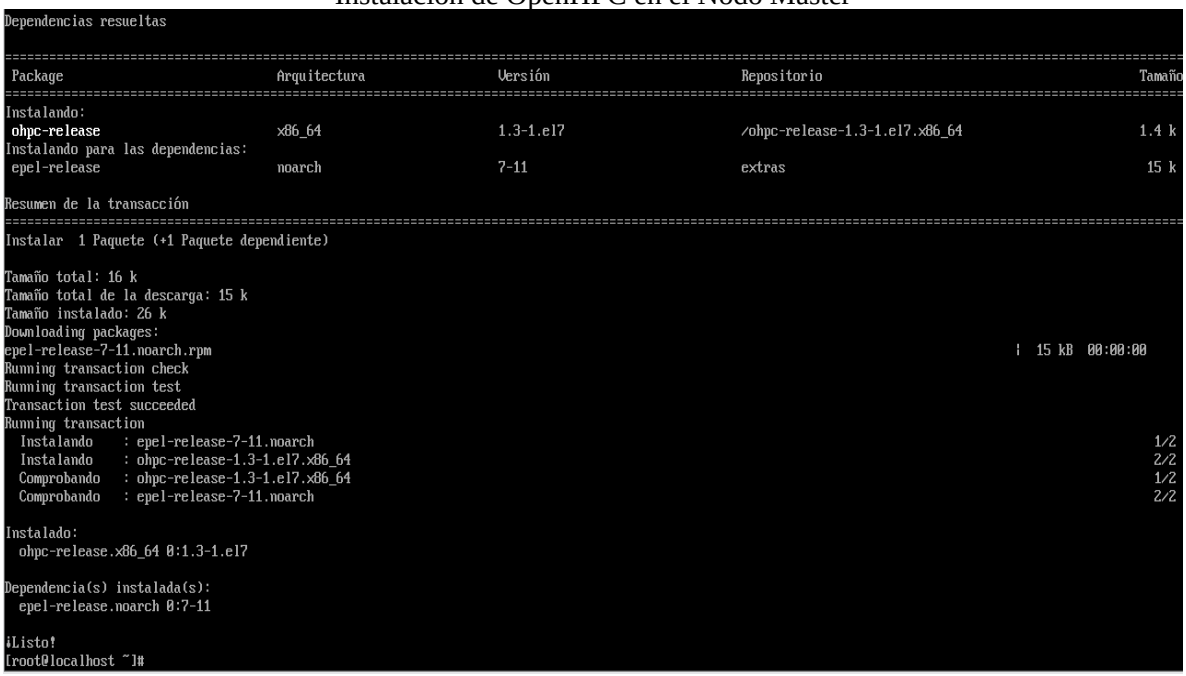


Fuente: El Autor

Se procede con la instalación del repositorio de OpenHPC que requiere acceso al nodo master por medio del siguiente comando y su proceso completado que se puede observar en el siguiente Gráfico 88.

Se instala OpenHPC en el nodo master
[root@localhost ~]# yum -y install
http://build.openhpc.community/OpenHPC:/1.3/CentOS_7/x86_64/ohpc-release-1.31.el7.x86_64.rpm

Gráfico 88
Instalación de OpenHPC en el Nodo Master

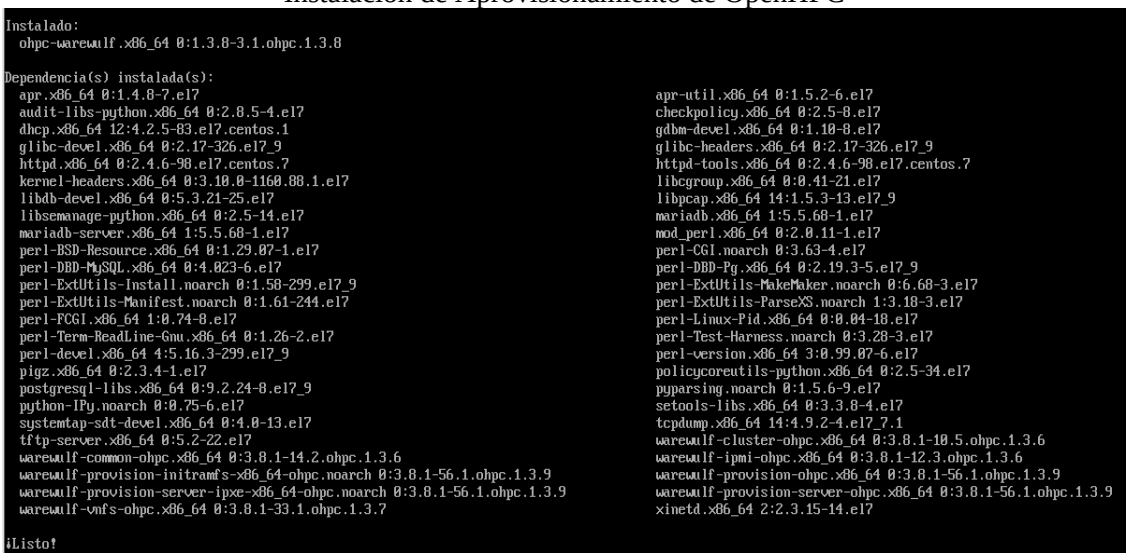


Fuente: El Autor

Se añade los servicios de aprovisionamiento en el nodo master, este repositorio provee una gama de componentes que servirán de ayuda para el proceso de configuración del clúster. Se lo realiza con los siguientes comandos y su instalación completa se muestra en el siguiente Gráfico 89.

Se instala el aprovisionamiento en el nodo master
[root@localhost ~]# yum -y install ohpc-base
[root@localhost ~]# yum -y install ohpc-warewulf

Gráfico 89
Instalación de Aprovisionamiento de OpenHPC



Fuente: El Autor

La plataforma HPC necesita sincronizar los relojes mediante el protocolo NTP (por sus siglas en ingles Network Time Protocol) en el enrutamiento de paquetes en la red, para ello se habilita el servicio NTP con el servidor time.haii.or.th en el host master con los siguientes comandos que se muestran en el siguiente Gráfico 90.

Se habilita el servicio NTP en el nodo master
[root@localhost ~]# systemctl enable ntpd.service
[root@localhost ~]# echo “server time.haii.or.th” >> /etc/ntp.conf
[root@localhost ~]# systemctl restart ntpd

Gráfico 90
Habilitación del Servicio NTP

Fuente: El Autor

Se añade los servicios de gestión de recursos en el nodo master, para ello se lo realiza por medio del servidor gestor de cargas de trabajo Slurm para luego añadir una imagen de cómputo mediante

los siguientes comandos que se pueden observar en el siguiente Gráfico 91.

Se añade los servicios de gestión de recursos en el nodo master.
[root@localhost ~]# yum -y install ohpc-slurm-server
[root@localhost ~]# perl -pi -e "s/ControlMachine=\S+/ControlMachine=master/" /etc/slurm/slurm.conf

Gráfico 91
Instalación de Gestión de Recursos

Fuente: El Autor

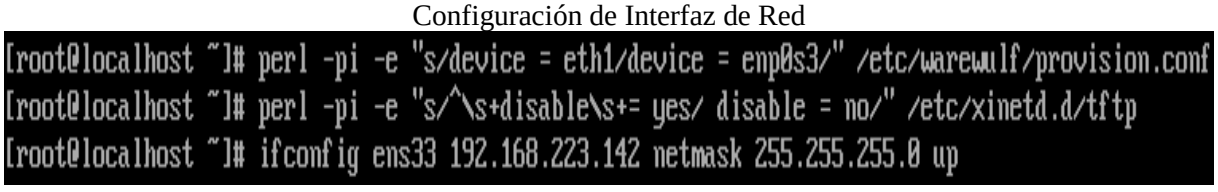
Se actualiza los archivos de configuración de Warewulf para que trabaje en CentOS 7, luego se configura la interfaz de red interna del nodo master y habilitarla para su aprovisionamiento como se detalla con los siguientes comandos en el siguiente Gráfico 92.

Configuración de interfaz interna.
[root@localhost ~]# perl -pi -e "s/device = eth1/device = ens33/" /etc/warewulf/provision.conf

Habilitar el servicio tftp para la distribución de imagen en el nodo de cómputo.
[root@localhost ~]# perl -pi -e "s/^\s+disable\s+= yes/ disable = no/" /etc/xinetd.d/tftp

Habilitar la interfaz interna para aprovisionamiento.
[root@localhost ~]# ifconfig ens33 192.168.223.142 netmask 255.255.255.0 up

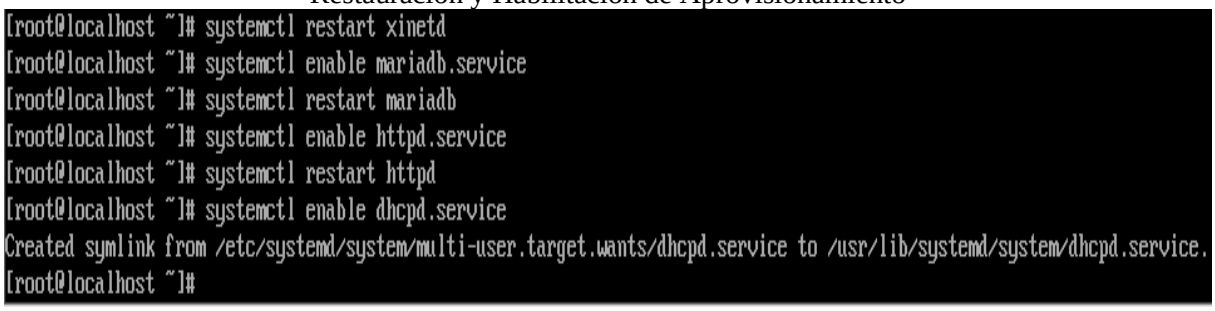
Gráfico 92
Configuración de Interfaz de Red



Fuente: El Autor

Se restaura y habilita los servicios de apoyo para aprovisionamiento mediante los siguientes comandos que se puede apreciar en el siguiente Gráfico 93.

Gráfico 93
Restauración y Habilitación de Aprovisionamiento



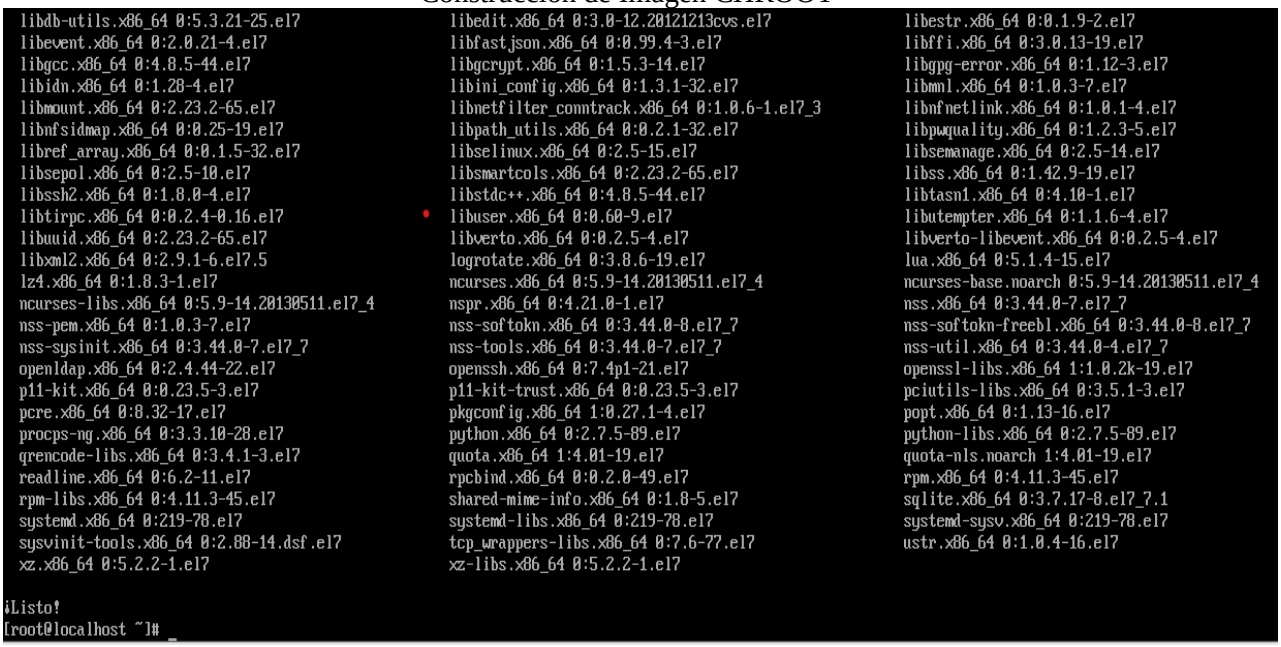
Fuente: El Autor

Con los siguientes comandos se define la locación de chroot la versión del CentOS 7.7 para acceder al repositorio externo mirror.centos.org durante el proceso de wwmkchroot, una vez finalizada la configuración se puede divisar en el siguiente Gráfico 94.

Define la locación chroot.
[root@localhost ~]# export CHROOT=/opt/ohpc/admin/images/centos7.7

Construcción inicial de la imagen chroot.
[root@localhost ~]# wwmkchroot centos-7 $CHROOT

Gráfico 94
Construcción de Imagen CHROOT

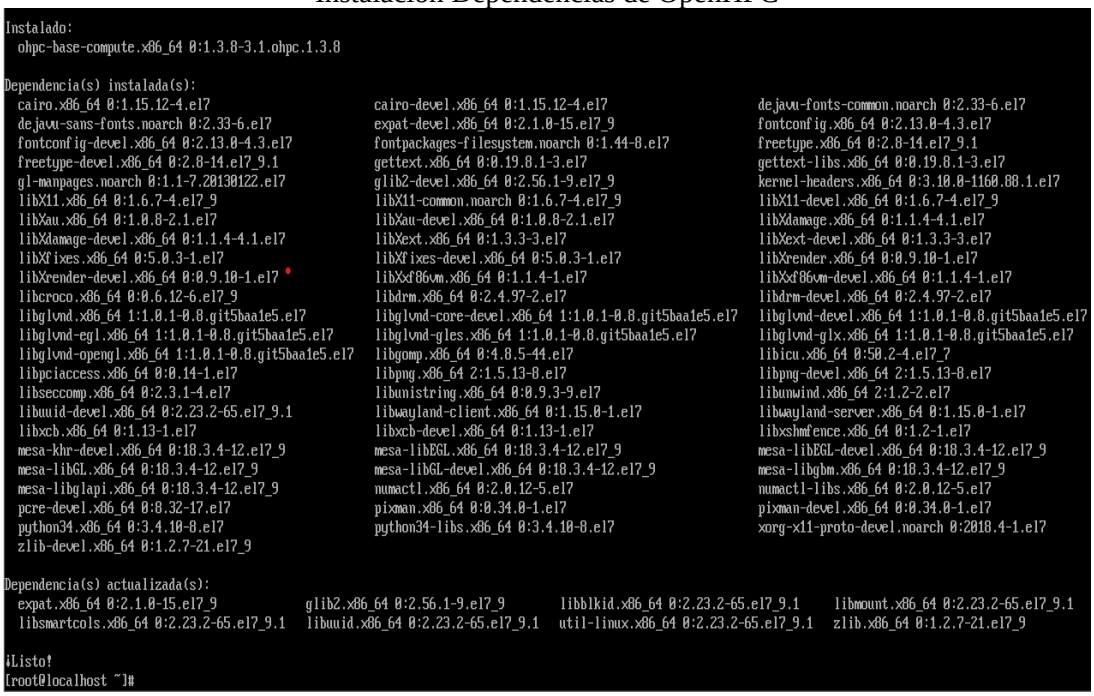


Fuente: El Autor

Se procede añadir los componentes de OpenHPC para que el cliente acceda a los recursos de gestión de servicios en el entorno de OpenHPC. La base de chroot instalada es ejecutada por wwmkchroot para modificar la base de aprovisionamiento y acceder a los repositorios, una vez finalizada la instalación se puede apreciar en el siguiente Gráfico 95.

Se instala la base meta-package en el nodo computo.
[root@localhost ~]# yum -y --installroot=$CHROOT install ohpc-base-compute

Gráfico 95
Instalación Dependencias de OpenHPC



Fuente: El Autor

Por medio del entorno chroot necesita ser actualizado para habilitar el DNS, para que el host master trabaje con la configuración del DNS se realiza una copia del archivo resolv.conf para el nodo master.

Se crea el archivo resolv.conf en el nodo computo.
[root@localhost ~]# cp -p /etc/resolv.conf $CHROOT/etc/resolv.conf

Se instalan los componentes adicionales al nodo master mediante distribución de paquetes yum en el usuario chroot, como se puede divisar en el siguiente Gráfico 96.

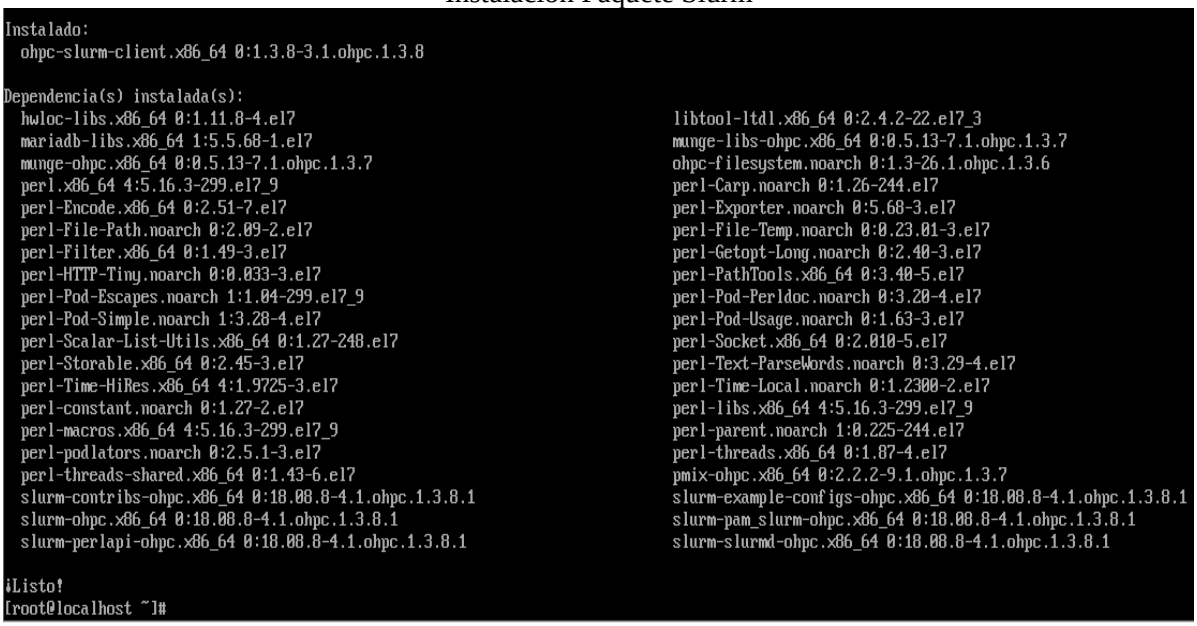
Se añade soporte cliente Slurm meta-package.
[root@localhost ~]# yum -y --installroot=$CHROOT install ohpc-slurm-client

Se añade el ntp (protocolo de tiempo de red).
[root@localhost ~]# yum -y --installroot=$CHROOT install ntp

Se añade drivers al kernel.
[root@localhost ~]# yum -y --installroot=$CHROOT install kernel

Se incluyen módulos en el entorno usuario.
[root@localhost ~]# yum -y --installroot=$CHROOT install lmod-ohpc

Gráfico 96
Instalación Paquete Slurm



Fuente: El Autor

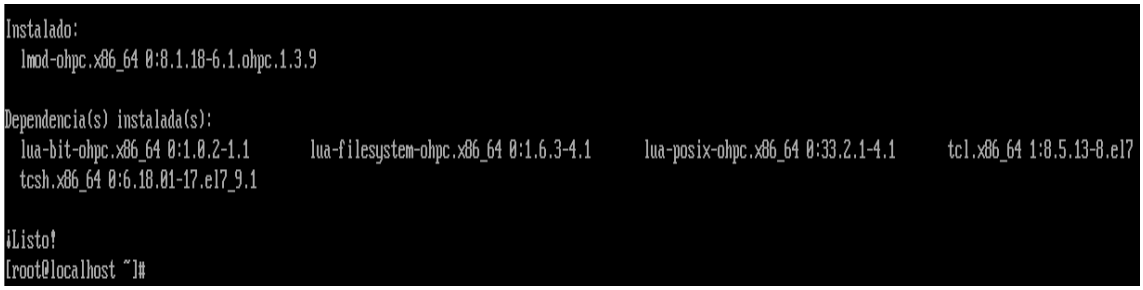
Se añade los drivers al kernel que facilita la comunicación entre los procesos y el hardware de los nodos, como se puede observar en el siguiente Gráfico 97.

Gráfico 97
Instalación del Kernel

Fuente: El Autor

Se instalan los módulos de usuario que permite la carga del software para ejecutarse en el clúster que se visualiza en el siguiente Gráfico 98.

Gráfico 98
Instalación de Módulos de Usuario

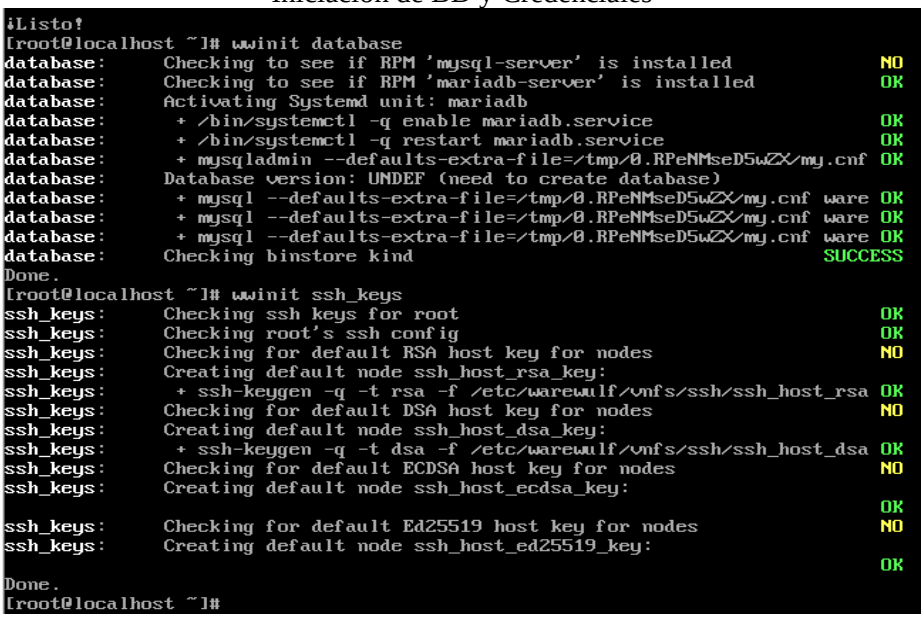


Fuente: El Autor

Se inicializa los servicios de la base de datos para añadir credenciales de acceso ssh por Warewulf al acceso remoto con los siguientes comandos que se pueden apreciar en el siguiente Gráfico 99.

Se inicializa base de datos Warewulf y accesos ssh.
[root@localhost ~]# wwinit database
[root@localhost ~]# wwinit ssh_keys

Gráfico 99
Iniciación de BD y Credenciales

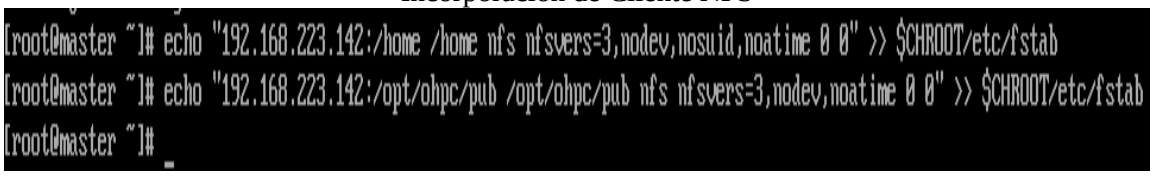


Fuente: El Autor

Se añade el cliente NFS (por sus siglas en ingles Network File System) que sirve para almacenar archivos en la red, como se observa en el siguiente Gráfico 100.

Se añade cliente NFS de /home y /opt/ohpc/pub a la imagen base
[root@localhost ~]# echo "192.168.223.142:/home /home nfs nfsvers=3,nodev,nosuid,noatime 0 0" >> $CHROOT/etc/fstab
[root@localhost ~]# echo "192.168.21.142:/opt/ohpc/pub /opt/ohpc/pub nfs nfsvers=3,nodev,noatime 0 0" >> $CHROOT/etc/fstab

Gráfico 100
Incorporación de Cliente NFS



Fuente: El Autor

Ahora se procede con crear el servidor NFS para que el usuario acceda a los archivos y directorios ubicados en el clúster OpenHPC, como se aprecia en el siguiente Gráfico 101.

Se crea NFS server para compartir el directorio de /home y /opt/ohpc/pub en el nodo master
[root@localhost ~]# echo "/home *(rw,no_subtree_check,fsid=10,no_root_squash)" >> /etc/exports
[root@localhost ~]# echo "/opt/ohpc/pub *(ro,no_subtree_check,fsid=11)" >> /etc/exports
[root@localhost ~]# exportfs -a
[root@localhost ~]# systemctl restart nfs-server
[root@localhost ~]# systemctl enable nfs-server

Gráfico 101
Creación del Servidor NFS

Fuente: El Autor

Se añade el servidor NTP en el nodo master para obtener información en tiempo real del clúster OpenHPC que se puede visualizar en el siguiente Gráfico 102.

Se añade servidor NTP del nodo master hacia los nodos cómputos
[root@localhost ~]# echo "server 192.168.223.142" >> $CHROOT/etc/ntp.conf
[root@localhost ~]# chroot $CHROOT systemctl enable ntpd

Gráfico 102
Adición de Servidor NTP

Fuente: El Autor

Se configura el script perl en los nodos c1 y c2 en los procesos del nombre, particiones, núcleos, etc; para luego habilitar e inicializar los servicios de munge en los demonios de ganglia y slurmctld para gestionarlos, lo que se muestra en el siguiente Gráfico 103.

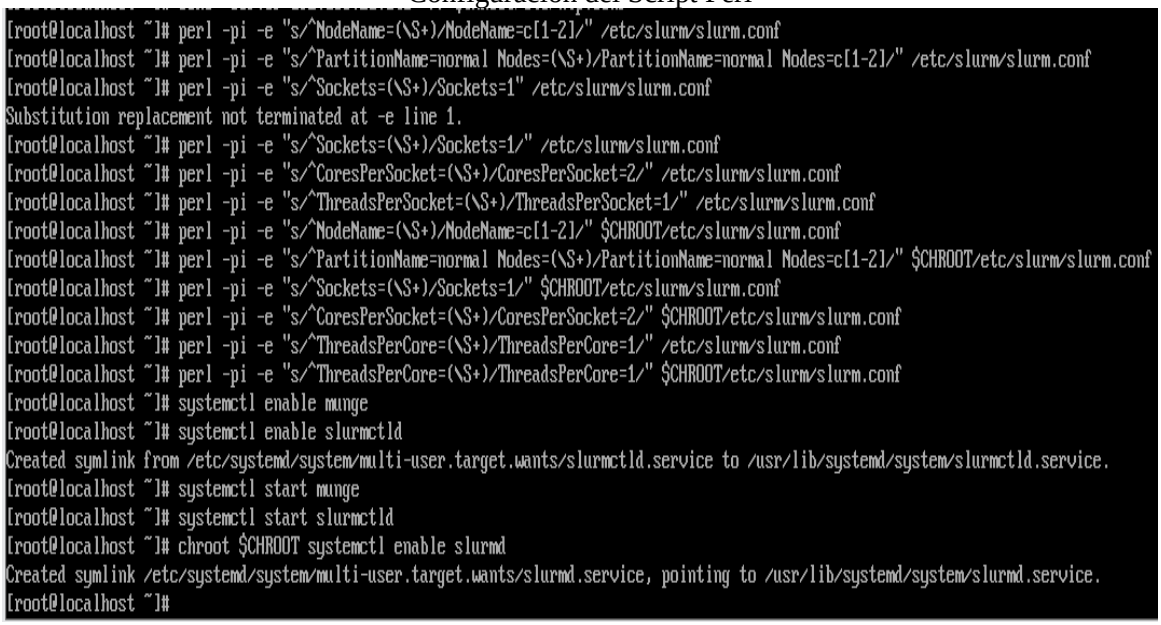
```
[root@localhost ~]# perl -pi -e "s/^NodeName=(\S+)/NodeName=c[1-2]/" /etc/slurm/slurm.conf
[root@localhost ~]# perl -pi -e "s/^PartitionName=normal Nodes=(\S+)/PartitionName=normal
Nodes=c[1-2]/" /etc/slurm/slurm.conf
[root@localhost ~]# perl -pi -e "s/^Sockets=(\S+)/Sockets=1/" /etc/slurm/slurm.conf
[root@localhost ~]# perl -pi -e "s/^CoresPerSocket=(\S+)/CoresPerSocket=2/"
/etc/slurm/slurm.conf
[root@localhost ~]#  perl -pi -e "s/^ThreadsPerCore=(\S+)/ThreadsPerCore=1/"
/etc/slurm/slurm.conf

[root@localhost ~]#  perl -pi -e "s/^NodeName=(\S+)/NodeName=c[1-2]/"
$CHROOT/etc/slurm/slurm.conf
[root@localhost ~]#   perl -pi -e "s/^PartitionName=normal Nodes=(\S+)/PartitionName=normal
Nodes=c[1-2]/" $CHROOT/etc/slurm/slurm.conf
[root@localhost ~]#  perl -pi -e "s/^Sockets=(\S+)/Sockets=1/" $CHROOT/etc/slurm/slurm.conf
[root@localhost ~]#  perl -pi -e "s/^CoresPerSocket=(\S+)/CoresPerSocket=2/"
$CHROOT/etc/slurm/slurm.conf
[root@localhost ~]#  perl -pi -e "s/^ThreadsPerCore=(\S+)/ThreadsPerCore=1/"
$CHROOT/etc/slurm/slurm.conf

[root@localhost ~]# systemctl enable munge
[root@localhost ~]# systemctl enable slurmctld
[root@localhost ~]# systemctl start munge
[root@localhost ~]# systemctl start slurmctld
[root@localhost ~]# chroot $CHROOT systemctl enable slurmd
```

Gráfico 103

Configuración del Script Perl



Fuente: El Autor

Se incrementa los límites de memoria con la finalidad de mejorar la velocidad de interconexión en las aplicaciones de los usuarios. Estos procesos que se presentan sirven para una mejor ejecución desde el host master hacia los nodos cómputos, también se habilita el módulo Slurm Pam como se puede apreciar en el siguiente Gráfico 104.

Se crea NFS server para compartir el directorio de /home y /opt/ohpc/pub en el nodo master

```
[root@localhost ~]# perl -pi -e 's/# End of file/\* soft memlock unlimited\n$&/s' /etc/security/limits.conf
[root@localhost ~]# perl -pi -e 's/# End of file/\* hard memlock unlimited\n$&/s' /etc/security/limits.conf
[root@localhost ~]# perl -pi -e 's/# End of file/\* soft memlock unlimited\n$&/s' $CHROOT/etc/security/limits.conf
[root@localhost ~]# perl -pi -e 's/# End of file/\* hard memlock unlimited\n$&/s' $CHROOT/etc/security/limits.conf
[root@localhost ~]# echo "account required pam_slurm.so" >> $CHROOT/etc/pam.d/sshd
```

Gráfico 104
Servidor NFS para Memoria

Fuente: El Autor

Se habilita el reenvío de registros del sistema para garantizar el envío de información para que el clúster provea el acceso fácil a los datos y reducir el impacto de almacenamiento en los nodos cómputos como se puede divisar en el siguiente Gráfico 105.

Se realiza la configuración para el nodo master

```
[root@localhost ~]# perl -pi -e "s/\\#\\\$ModLoad imudp/\\\$ModLoad imudp/" /etc/rsyslog.conf
[root@localhost ~]# perl -pi -e "s/\\#\\\$UDPServerRun 514/\\\$UDPServerRun 514/"
/etc/rsyslog.conf
[root@localhost ~]# systemctl restart rsyslog
[root@localhost ~]# echo "*.* @192.168.223.142:514" >> $CHROOT/etc/rsyslog.conf
[root@localhost ~]# perl -pi -e "s/^\*\.info/\\#\*\.info/" $CHROOT/etc/rsyslog.conf
[root@localhost ~]# perl -pi -e "s/^authpriv/\\#authpriv/" $CHROOT/etc/rsyslog.conf
[root@localhost ~]# perl -pi -e "s/^mail/\\#mail/" $CHROOT/etc/rsyslog.conf
[root@localhost ~]# perl -pi -e "s/^cron/\\#cron/" $CHROOT/etc/rsyslog.conf
[root@localhost ~]# perl -pi -e "s/^uucp/\\#uucp/" $CHROOT/etc/rsyslog.conf
```

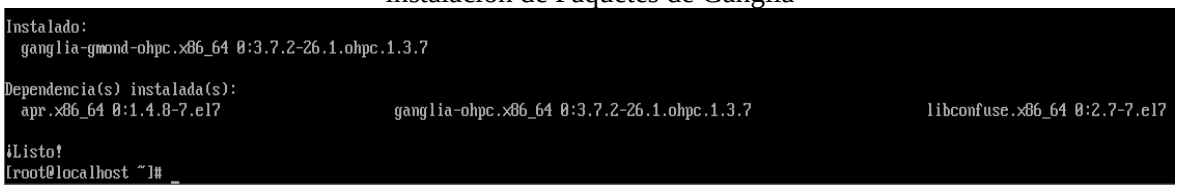
Gráfico 105
Habilitación de Registros del Sistema

Fuente: El Autor

Se procede con la instalación del paquete de Ganglia que monitorea el desempeño del clúster en el nodo master que se aprecia en el siguiente Gráfico 106.

Se instala Ganglia meta-package en el nodo master
[root@localhost ~]# yum -y install ohpc-ganglia
Se instala Ganglia en los nodos computo
[root@localhost ~]# yum -y --installroot=$CHROOT install ganglia-gmond-ohpc

Gráfico 106
instalación de Paquetes de Ganglia



Fuente: El Autor

Se instala el demonio gmond de OpenHPC de ganglia en el nodo master que se divisa en el siguiente Gráfico 107.

Gráfico 107
Instalación de los Demonios de Ganglia

Fuente: El Autor

Se configura el receptor del demonio gmond de ganglia, luego se añade el nombre del grid que es del TesisMaster y se habilita los servicios de los demonios de ganglia para ultimo ingresar desde el navegador web con la ip publica https://localhost.com/ganglia que se aprecia en el siguiente Gráfico 108.

Uso de script de configuración para habilitar receptor único del nodo master
[root@localhost ~]# cp /opt/ohpc/pub/examples/ganglia/gmond.conf /etc/ganglia/gmond.conf
[root@localhost ~]# perl -pi -e "s/<sms>/master/" /etc/ganglia/gmond.conf
Se añade configuración al nodo cómputo y se coloca el nombre del grid
[root@localhost ~]# cp /etc/ganglia/gmond.conf $CHROOT/etc/ganglia/gmond.conf
[root@localhost ~]# echo "gridname TesisMaster..." >> /etc/ganglia/gmetad.conf

Se comienza habilitar los servicios de ganglia
[root@localhost ~]# systemctl enable gmond
[root@localhost ~]# systemctl enable gmetad
[root@localhost ~]# systemctl start gmond
[root@localhost ~]# systemctl start gmetad
[root@localhost ~]# chroot $CHROOT systemctl enable gmond
Se restaura el servidor web
[root@localhost ~]# systemctl try-restart httpd

Gráfico 108
Habilitación de los Servicios de Ganglia

Fuente: El Autor

Se añade el ClusterShell basado en librería de Python para poder ejecutar comandos in paralelo a través de los nodos del clúster. Esta configuración define el grid de tres nodos que son el administrador y los nodos cómputos como se puede visualizar en el siguiente Gráfico 109.

Se instala ClusterShell
[root@localhost ~]# yum -y install clustershell-ohpc
Se configura la definición del nodo
[root@localhost ~]# cd /etc/clustershell/groups.d
[root@localhost ~]# mv local.cfg local.cfg.orig
[root@localhost ~]# echo "adm: master" > local.cfg
[root@localhost ~]# echo "compute: c [1-2]" >> local.cfg
[root@localhost ~]# echo "all: @adm,@compute" >> local.cfg

Gráfico 109
Instalación de Clustershell

```
[root@localhost ~]# yum -y install clustershell-ohpc
Complementos cargados:fastestmirror
Loading mirror speeds from cached hostfile
 * base: centos.ufms.br
 * epel: ftp-chi.osuosl.org
 * extras: centos.ufms.br
 * updates: centos.ufms.br
El paquete clustershell-ohpc-1.8.2-3.1.ohpc.1.3.9.noarch ya se encuentra instalado con su versión más reciente
Nada para hacer
[root@localhost ~]# cd /etc/clustershell/groups.d
[root@localhost groups.d]# mv local.cfg local.cfg.orig
mv: ¿sobreescribir «local.cfg.orig»? (s/n) s
[root@localhost groups.d]# echo "adm: master" > local.cfg
[root@localhost groups.d]# echo "compute: c[1-2]" >> local.cfg
[root@localhost groups.d]# echo "all: @adm,@compute" >> local.cfg
[root@localhost groups.d]# cd
[root@localhost ~]#
```

Fuente: El Autor

Se incluyen los archivos de importación para distribuir las credenciales de usuario a los nodos cómputos como se puede apreciar con los comandos en el siguiente Gráfico 110.

Se añade archivo de configuración de Slurm y credencial criptográfica para la autenticación.
[root@localhost ~]# wwsh file import /etc/passwd
[root@localhost ~]# wwsh file import /etc/group
[root@localhost ~]# wwsh file import /etc/shadow
[root@localhost ~]# wwsh file import /etc/slurm/slurm.conf
[root@localhost ~]# wwsh file import /etc/munge/munge.key

Gráfico 110
Archivos de Credenciales de Usuario

```
[root@localhost ~]# wwsh file import /etc/passwd
[root@localhost ~]# wwsh file import /etc/group
[root@localhost ~]# wwsh file import /etc/shadow
[root@localhost ~]# wwsh file import /etc/slurm/slurm.conf
[root@localhost ~]# wwsh file import /etc/munge/munge.key
[root@localhost ~]# wwsh file list
group                :  rw-r--r-- 1   root root             684 /etc/group
munge.key            :  r-------- 1   munge munge          1024 /etc/munge/munge.key
passwd               :  rw-r--r-- 1   root root            1628 /etc/passwd
shadow               :  rw-r----- 1   root root             982 /etc/shadow
slurm.conf           :  rw-r--r-- 1   root root            2148 /etc/slurm/slurm.conf
[root@localhost ~]#
```

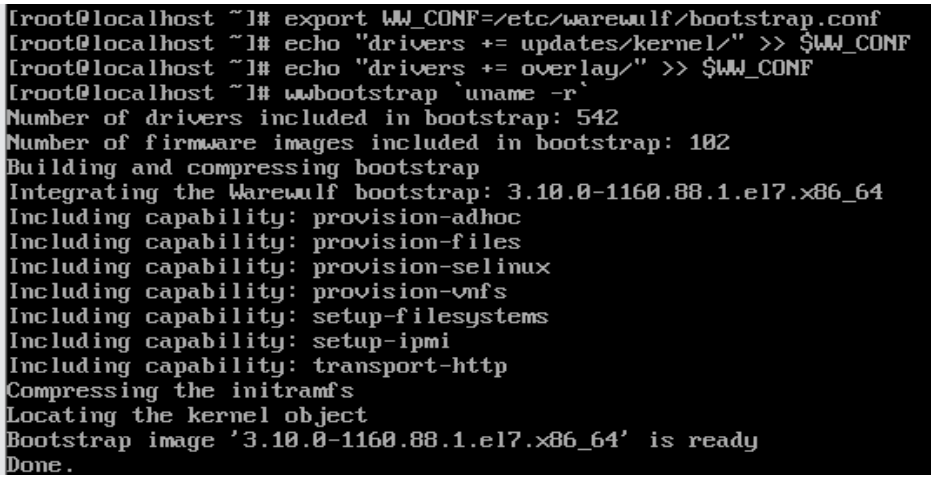
Fuente: El Autor

Se añade una imagen Bootstrap para el kernel en tiempo de ejecución con procesos de aprovisionamiento por medio de scripts simples, se requieren incluir drivers para la creación del

Bootstrap como se puede divisar en el siguiente Gráfico 111.

Se actualiza drivers desde el kernel.
[root@localhost ~]# export WW_CONF=/etc/warewulf/bootstrap.conf
[root@localhost ~]# echo "drivers += updates/kernel/" >> $WW_CONF
Se incluyen drivers overlay.
[root@localhost ~]# echo "drivers += overlay" >> $WW_CONF
Se construye la imagen bootstrap.
[root@localhost ~]# wwbootstrap `uname -r`

Gráfico 111
Actualización del Kernel

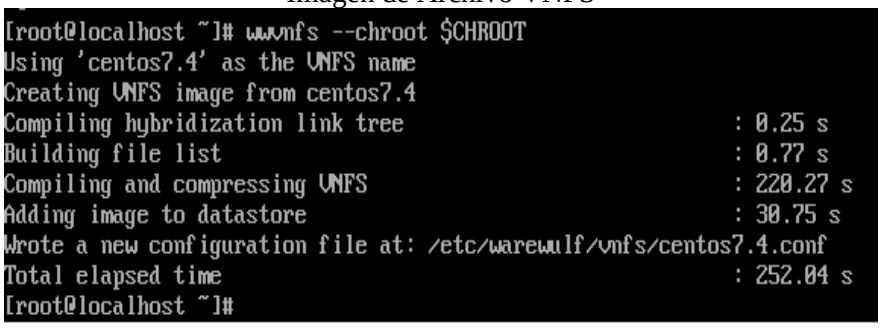


Fuente: El Autor

Se arma una imagen de sistema de archivo de nodo virtual (VNFS por sus siglas en ingles Virtual Node File System) para encapsular el entorno chroot como se aprecia en el siguiente Gráfico 112.

Se añade imagen de sistema de nodo virtual.
[root@localhost ~]# wwvnfs –chroot $CHROOT

Gráfico 112
Imagen de Archivo VNFS

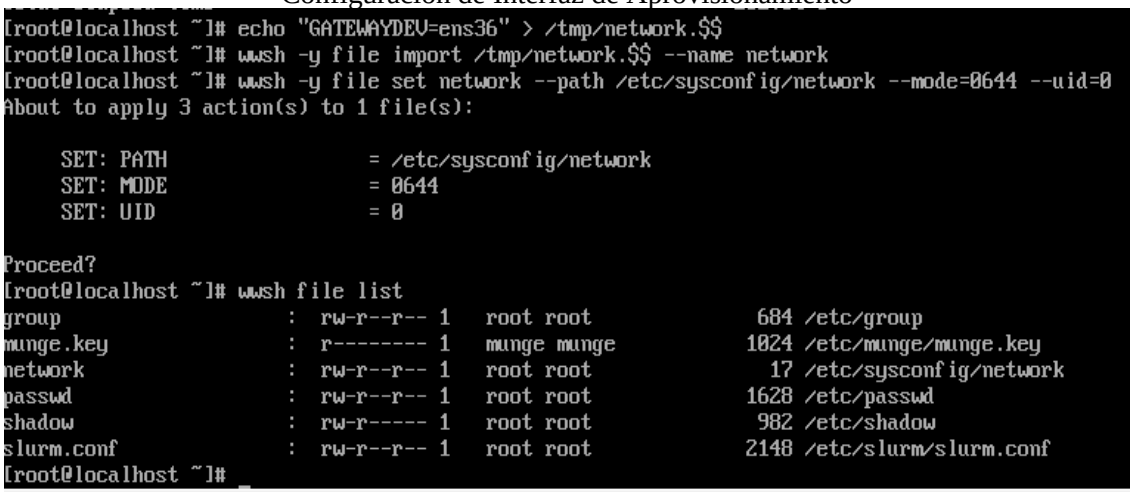


Fuente: El Autor

Se registran los archivos en los nodos de aprovisionamiento que define el clúster a través del dispositivo de traducción de información GATEWATDEV que se observa en el siguiente Gráfico 113.

Se configura la interfaz de aprovisionamiento como dispositivo de red por defecto.
[root@localhost ~]# echo "GATEWAYDEV=ens36" > /tmp/network.$$
[root@localhost ~]# wwsh -y file import /tmp/network.$$ --name network
[root@localhost ~]# wwsh -y file set network --path /etc/sysconfig/network --mode=0644 --uid=0

Gráfico 113
Configuración de Interfaz de Aprovisionamiento

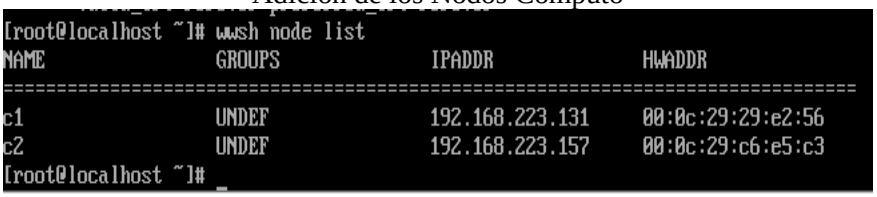


Fuente: El Autor

Se añade cada uno de los nodos cómputo C1 y C2 con sus respectivas direcciones ip y mac para que se registren en el grid TesisMaster que se observa en el siguiente Gráfico 114.

Se añaden los nodos al almacenamiento de datos Warewulf.
[root@localhost ~]# wwsh -y node new c1 --ipaddr=192.168.23.131 --hwaddr=00:0C:29:29:E2:56 -D ens36
[root@localhost ~]# wwsh -y node new c2 --ipaddr=192.168.223.157 --hwaddr=00:0C:29:C6:E5:C3 -D ens36
[root@localhost ~]# wwsh node list

Gráfico 114
Adición de los Nodos Computo

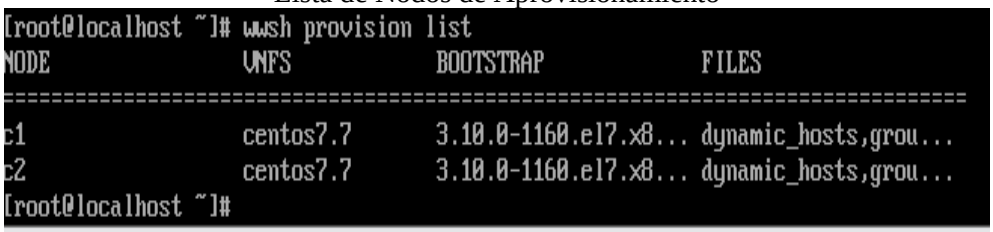


Fuente: El Autor

Se define la imagen VNFS, se restaura los servicios de ganglia y dhcp con la finalidad de visualizar el grid TesisMaster en el navegador web como se aprecia en el siguiente Gráfico 115.

Se define la imagen VNFS.
[root@localhost ~]# wwsh -y provision set "c1" --vnfs=centos7.7 --bootstrap=`uname -r` --files=dynamic_hosts,passwd,group,shadow,slurm.conf,munge.key,network
[root@localhost ~]# wwsh -y provision set "c2" --vnfs=centos7.7 --bootstrap=`uname -r` --files=dynamic_hosts,passwd,group,shadow,slurm.conf,munge.key,network
Se restaura los servicios de ganglia/dhcp.
[root@localhost ~]# systemctl restart gmond
[root@localhost ~]# systemctl restart gmetad
[root@localhost ~]# systemctl restart dhcpd

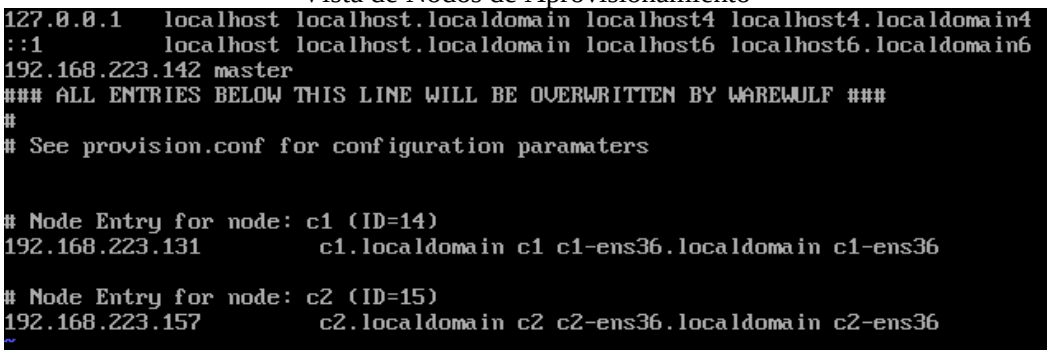
Gráfico 115
Lista de Nodos de Aprovisionamiento



Fuente: El Autor

Se verifica los servicios de aprovisionamientos de los nodos registrados en el directorio /etc/hosts que se encuentran en el nodo master como se puede visualizar en el siguiente Gráfico 116.

Gráfico 116
Vista de Nodos de Aprovisionamiento

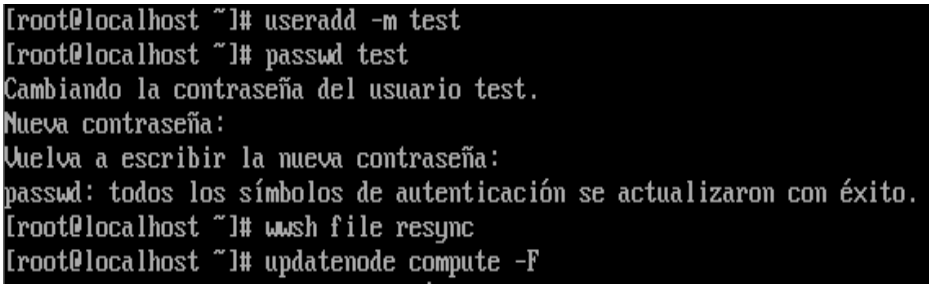


Fuente: El Autor

Se crea un usuario de prueba para habilitar trabajos de ejecución en el host master para sincronizar automáticamente los nodos cómputos como se muestra en el siguiente Gráfico 117.

Se crea un usuario de prueba.
[root@localhost ~]# useradd -m test
[root@localhost ~]# passwd test
[root@localhost ~]# wwsh file resync
[root@localhost ~]# updatenode compute -F

Gráfico 117
Creación de Usuario de Prueba

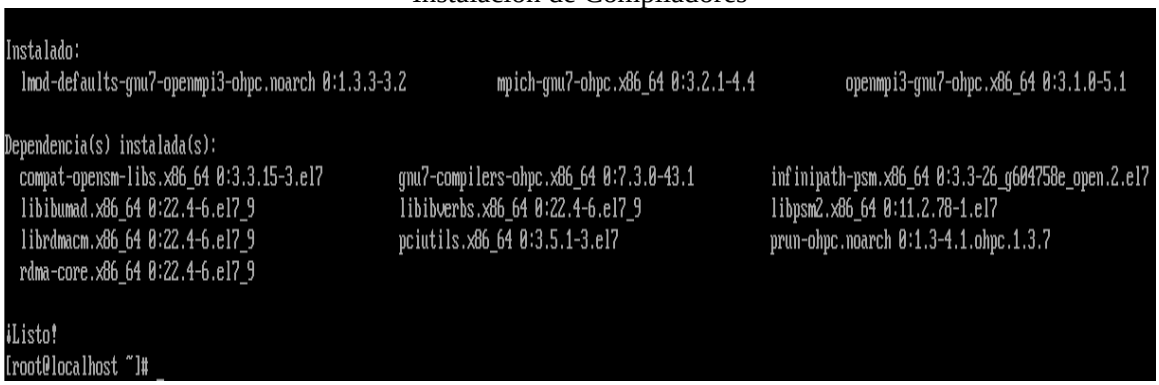


Fuente: El Autor

Se procede con la instalación de los compiladores de desarrollo MPICC que provee OpenHPC por medio del siguiente comando que se puede ver finalizada su instalación en el siguiente Gráfico 118.

Se instalan compiladores mpicc.
[root@localhost ~]# yum -y install openmpi3-gnu7-ohpc mpich-gnu7-ohpc lmod-defaults-gnu7-openmpi3-ohpc

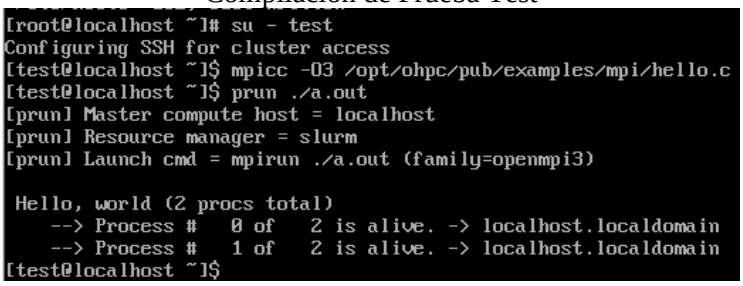
Gráfico 118
Instalación de Compiladores



Fuente: El Autor

Se procede con la compilación de prueba Hello World que demuestra los procesos enviados por el nodo master hacia los otros nodos. Se accede al clúster en modo superusuario a la prueba que se llama ***test***. Como se observa en el siguiente Gráfico 119 mediante los siguientes comandos se realiza el envío de procesos desde host master que se instaló en el proceso de configuración del clúster OpenHPC.

Gráfico 119
Compilación de Prueba Test



Fuente: El Autor

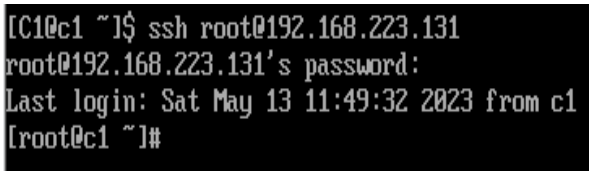
ANEXO 3. Configuración de los nodos C1 y C2

En el presente apartado de ANEXO 3 se procede con la configuración de los nodos C1 y C2 para visualizar el desempeño del clúster en la plataforma de Ganglia. El procedimiento de configuración es de igual manera para los nodos C1 y C2.

Se realiza la conexión vía SSH para comenzar con la configuración de los nodos como se observa en el siguiente Gráfico 120.

Se realiza la conexión vía ssh.
[C1@c1]# ssh root@192.168.223.131

Gráfico 120
Acceso SSH al nodo C1

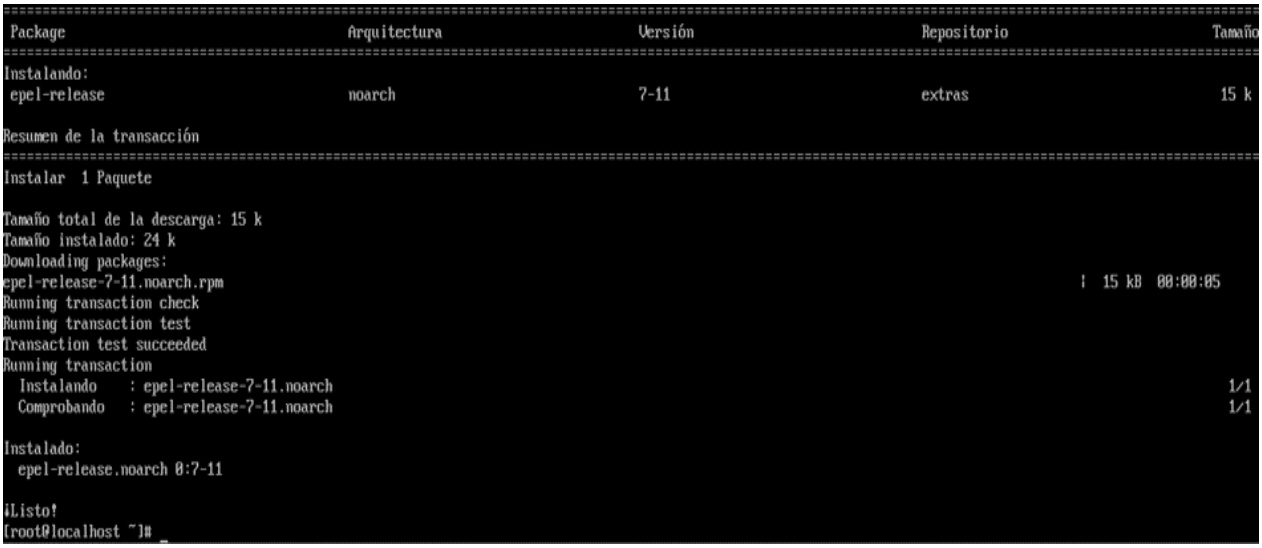


Fuente: El Autor

Se procede con la actualización del sistema operativo CentOS 7 con los siguientes comandos que se observa en el siguiente Gráfico 121.

Actualización de CentOS 7.
[root@c1 ~]# yum -y update
[root@c1 ~]# yum -y install epel-release

Gráfico 121
Actualización del SO en C1

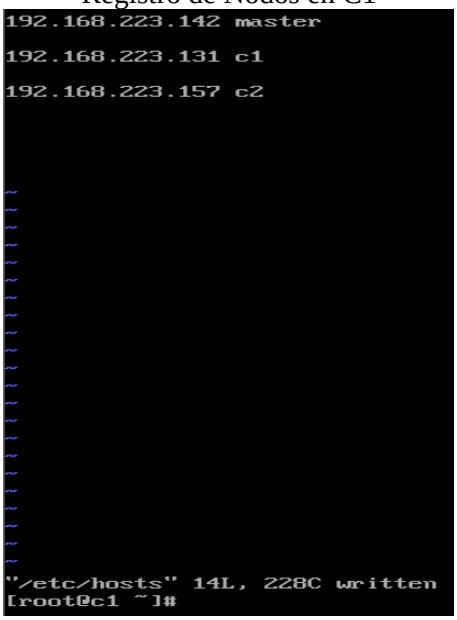


Fuente: El Autor

Se registra los nodos C1, C2 y master en el directorio de hosts, luego para finalizar se digita :wq para guardar los cambios respectivos que se observa en el siguiente Gráfico 122.

Registro de los nodos en el directorio de hosts.
[root@c1 ~]# vi /etc/hosts

Gráfico 122
Registro de Nodos en C1



Fuente: El Autor

Se deshabilita el firewall en los nodos con los siguientes comandos que se muestran a continuación en el Gráfico 123.

Deshabilita el firewall.
[root@c1 ~]# systemctl stop firewalld
[root@c1 ~]# systemctl disable firewalld

Gráfico 123
Deshabilitación de Firewall en C1

Fuente: El Autor

Se instala el demonio gmond de Ganglia en cada uno de los nodos C1 y C2 con el siguiente comando que se observa en el siguiente Gráfico 124.

Se instala el demonio de gmond.
[root@c1 ~]# yum install ganglia-gmond

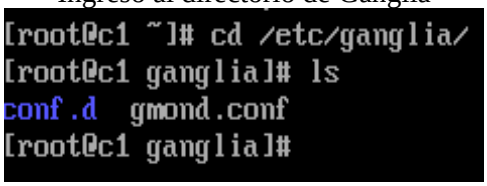
Gráfico 124
Instalación Demonio Gmond en C1

Fuente: El Autor

Se ingresa al directorio de ganglia para luego acceder a la configuración del demonio gmond por medio del siguiente comando que se aprecia en el siguiente Gráfico 125.

Acceso al directorio de ganglia.
[root@c1 ~]# cd /etc/ganglia/

Gráfico 125
Ingreso al directorio de Ganglia



Fuente: El Autor

Se realiza el ingreso a la configuración del demonio gmond para realizar los cambios en los parámetros en el nombre del clúster que en este caso es el de ***OpenHPC***, el udp_send_channel, el udp_recv_channel y el tcp_accept_channel. Lo cual debe quedar de la siguiente manera como se muestra en los siguientes Gráficos 126, 127 y 128. Se digita el comando :wq para guardar los cambios respectivos.

Acceso a la configuración del demonio gmond.
[root@c1 ganglia]# vi gmond.conf

Gráfico 126
Nombre del Clúster

Fuente: El Autor

Se comenta con el signo # los parámetros de bind_hostname y mcast_join para que solo quede habilitado los parámetros de host, port y ttl como se observa en el siguiente Gráfico 127.

Gráfico 127

Configuración Parámetro udp_send_channel

```
/* Feel free to specify as many udp_send_channels as you like.  Gmond
   used to only support having a single channel */
udp_send_channel {
  #bind_hostname = yes # Highly recommended, soon to be default.
                       # This option tells gmond to use a source address
                       # that resolves to the machine's hostname.  Without
                       # this, the metrics may appear to come from any
                       # interface and the DNS names associated with
                       # those IPs will be used to create the RRDs.
  # mcast_join = 239.2.11.71
  host = master
  port = 8649
  ttl = 1
}
```

Fuente: El Autor

Se comenta con el signo # los parámetros de mcast_join, bind, port, retry_bind, etc como se visualiza en el siguiente Gráfico 128.

Gráfico 128

Configuración Parámetro udp_recv_channel

```
/* You can specify as many udp_recv_channels as you like as well. */
udp_recv_channel {
  # mcast_join = 239.2.11.71
  # port = 8649
  # bind = 239.2.11.71
  # retry_bind = true
  # Size of the UDP buffer. If you are handling lots of metrics you really
  # should bump it up to e.g. 10MB or even higher.
  # buffer = 10485760
}
```

Fuente: El Autor

Se comenta con el signo # los parámetros de port, gzip XML y gzip_output como se aprecia en el siguiente Gráfico 129.

Gráfico 129

Configuración Parámetro tcp_accept_channel

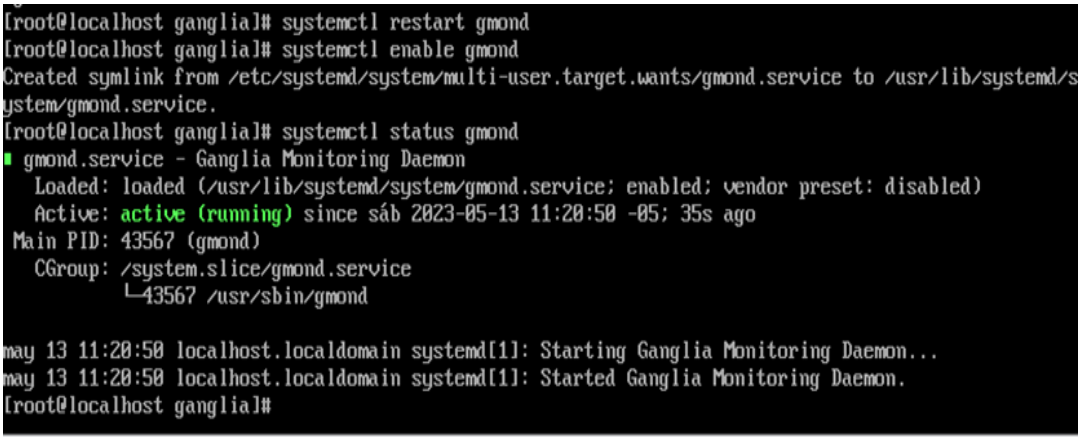
```
/* You can specify as many tcp_accept_channels as you like to share
   an xml description of the state of the cluster */
tcp_accept_channel {
  # port = 8649
  # If you want to gzip XML output
  # gzip_output = no
}
```

Fuente: El Autor

Se habilita el demonio gmond para luego verificar el estado del demonio que se ejecutando de manera correcta como se muestra en el siguiente Gráfico 130.

Se habilita el demonio gmond.
[root@c1 ganglia]# systemctl restart gmond
[root@c1 ganglia]# systemctl enable gmond
[root@c1 ganglia]# systemctl status gmond

Gráfico 130
Activación Demonio Gmond en C1



Fuente: El Autor

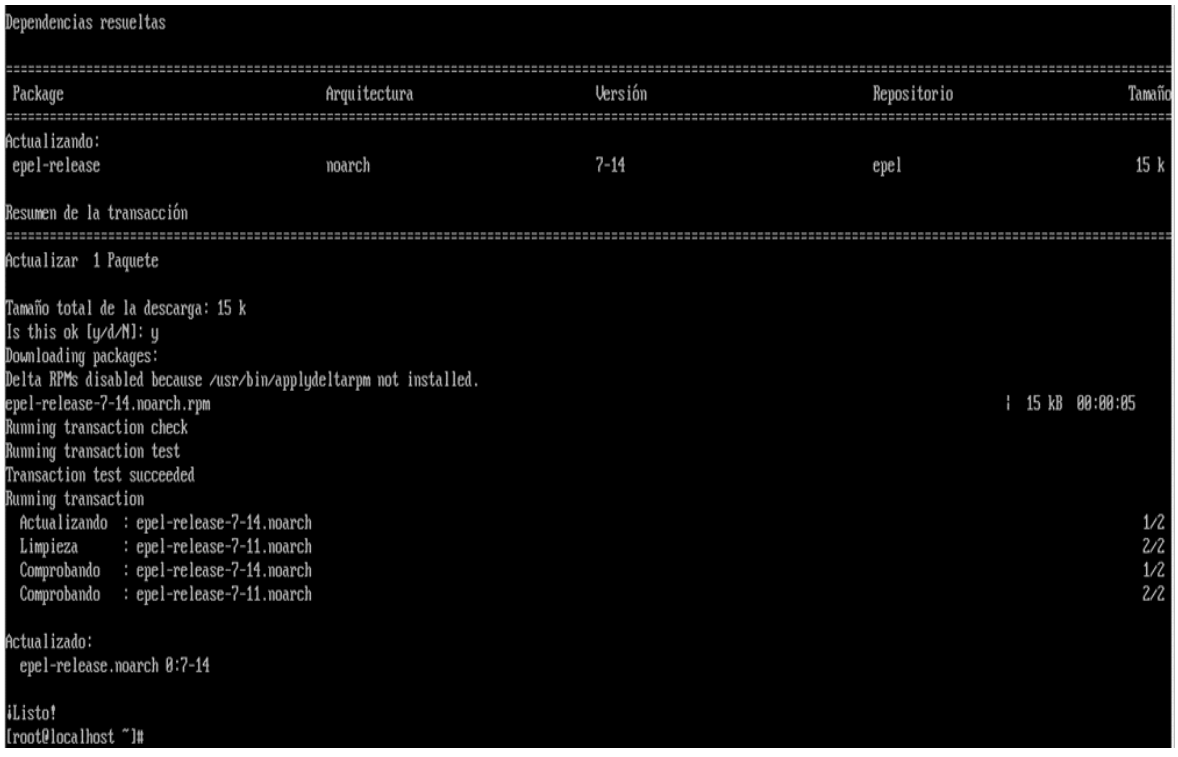
ANEXO 4. Instalación y Configuración de RStudio Server

Para el presente apartado del ANEXO 4 se procede con la instalación y configuración de RStudio Server en el nodo master que servirá como entorno de ejecución para el script de aproximación del número pi desarrollado en lenguaje R. A continuación, se presente los siguientes procesos con los respectivos comandos de instalación.

Previo a la instalación de R se recomienda proceder con la actualización del sistema operativo para que obtenga los paquetes más recientes para una interacción con el IDE de RStudio mediante el siguiente comando mostrado a continuación en el Gráfico 131.

Se actualiza el sistema operativo CentOS 7.
[root@localhost ~]# yum install epel-release

Gráfico 131
Actualización SO CentOS 7

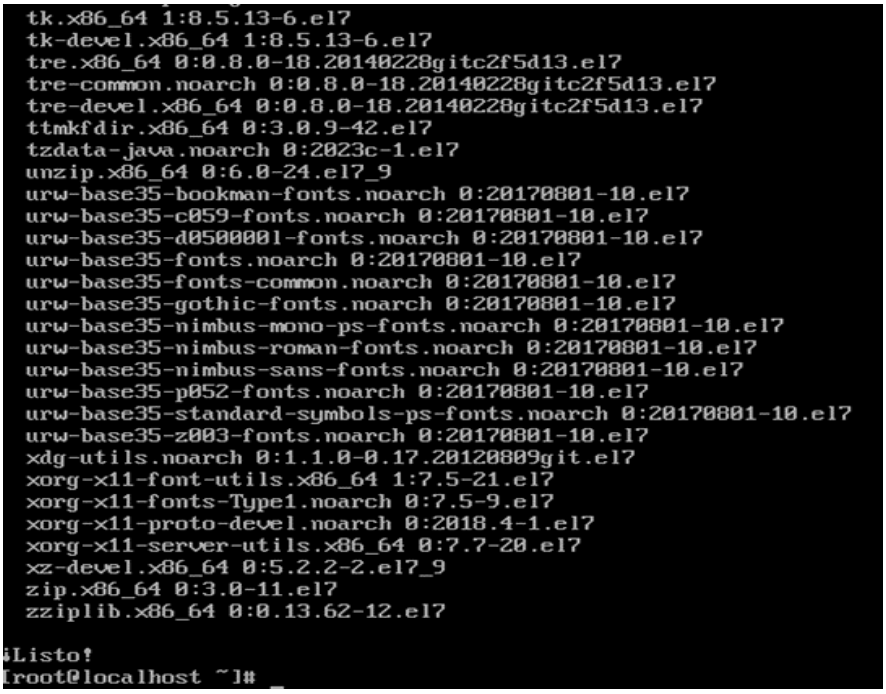


Fuente: El Autor

Se realiza la correspondiente instalación del lenguaje R por medio del respectivo comando que se muestra a continuación en el siguiente Gráfico 132.

Se instala el lenguaje R en el nodo master.
[root@localhost ~]# yum install R -y

Gráfico 132
Instalación de R



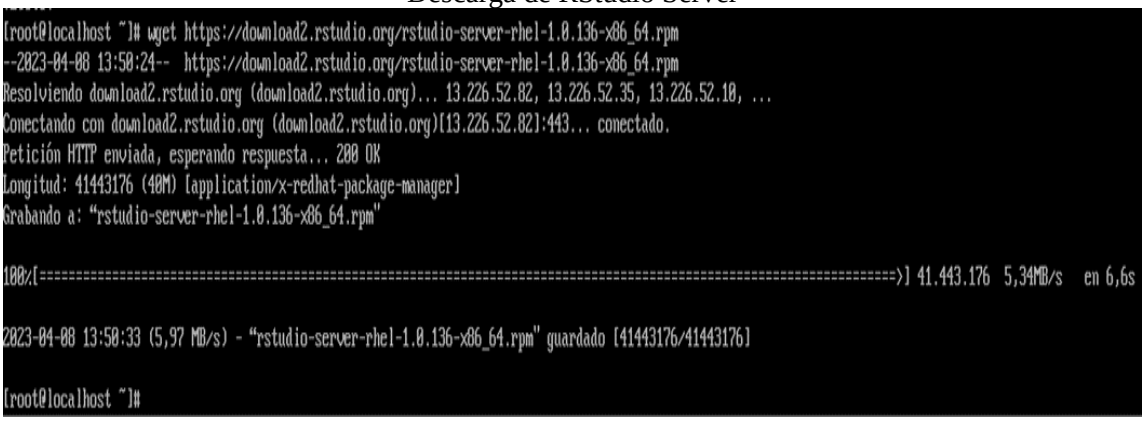
Fuente: El Autor

Se procede con la descarga del IDE RStudio Server para facilitar una mejor usabilidad del entono más interactivo con el siguiente comando que se muestra a continuación en el siguiente Gráfico 133.

Se descarga el entorno RStudio Server en el nodo master.
[root@localhost ~]# wget https://download2.rstudio.org/rstudio-server-rhel-1.0.136-x86_64.rpm

Gráfico 133

Descarga de RStudio Server



Fuente: El Autor

Se realiza con la instalación de RStudio Server en el nodo master con el correspondiente comando que se muestra a continuación en el siguiente Gráfico 134.

Se instala el entorno RStudio Server en el nodo master.
[root@localhost ~]# yum install –nogpgcheck rstudio-server-rhel-1.0.136-x86_64.rpm -y

Gráfico 134

Instalación de RStudio Server

Fuente: El Autor

Se comprueba que el estado del servicio de RStudio server mediante el siguiente comando que puede visualizarse en el siguiente Gráfico 135.

Se comprueba RStudio Server.
[root@localhost ~]# systemctl status rstudio-server.service

Gráfico 135
Estado de RStudio Server

Fuente: El Autor

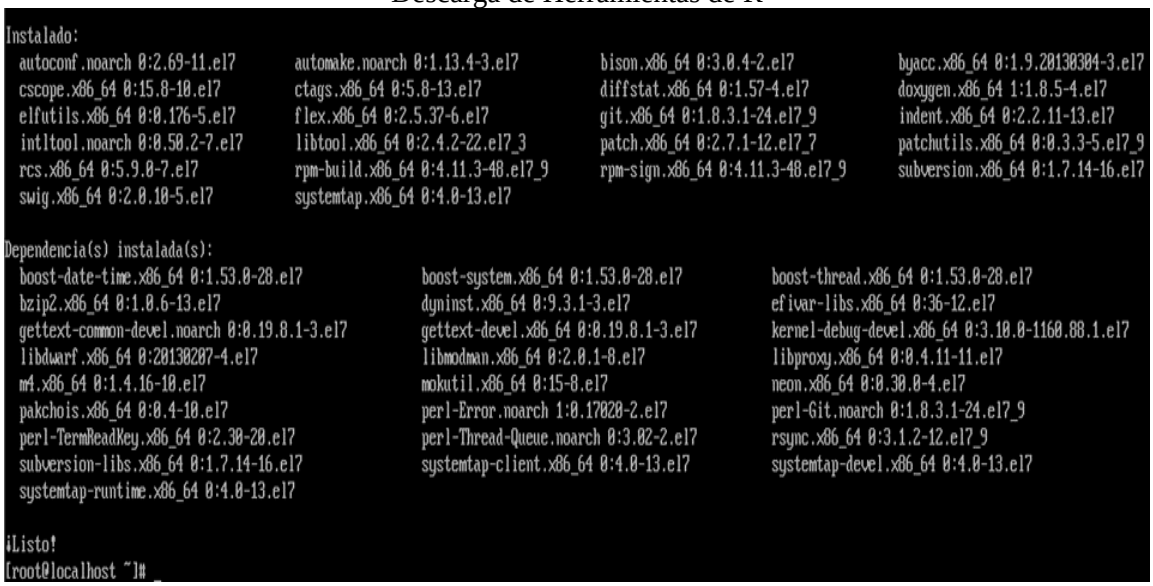
Nota: En caso de que el servicio de RStudio server no es activado se procede para habilitarlo junto con la apertura del puerto 8787 que es del protocolo de control de transmisión TCP mediante los siguientes comandos.

Se habilita RStudio Server.
[root@localhost ~]# systemctl enable rstudio-server.service
[root@localhost ~]# firewallcmd –permanent --zone=public --add-port=8787/tcp && firewall-cmd --reload

Se realiza la descarga del paquete de herramientas de desarrollo para R con el siguiente comando que se muestra a continuación en el Gráfico 136.

Descarga de herramientas de desarrollo.
[root@localhost ~]# yum groupinstall “Development Tools” -y

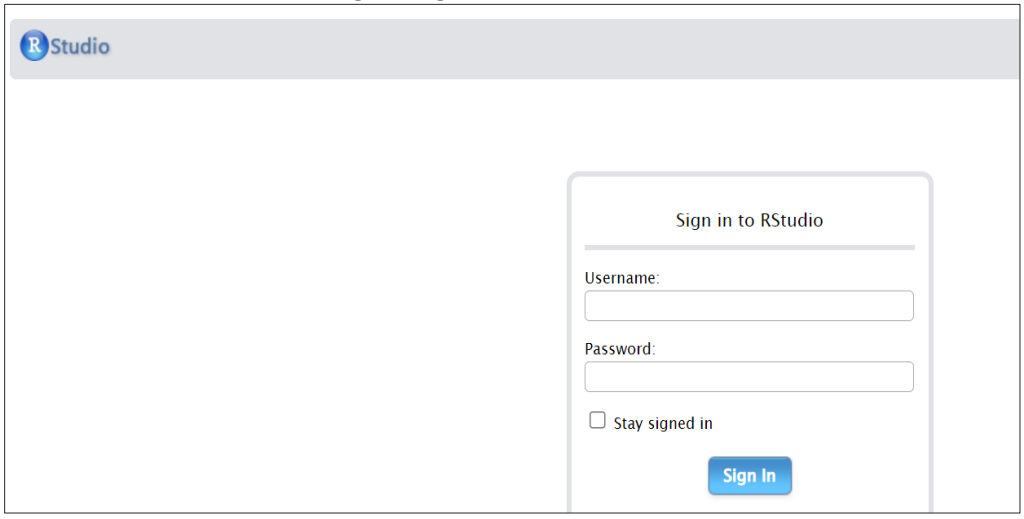
Gráfico 136
Descarga de Herramientas de R



Fuente: El Autor

Para acceder a RStudio Server desde el nodo master se digita en el navegador web con la dirección https://192.168.223.142:8787. Se digita las credenciales de username y password para acceder al entorno de RStudio Server como se muestra a continuación en el siguiente Gráfico 137.

Gráfico 137
Página Ingreso a RStudio Server



Fuente: El Autor