UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

Configuración y puesta en operación de un Clúster HPC para la Unidad MOFABI de la Facultad de Ingeniería de la UNAM

PROYECTO FINAL

QUE PARA OPTAR POR EL GRADO DE:

ESPECIALISTA EN CÓMPUTO DE ALTO RENDIMIENTO

P R E S E N T A:

LOURDES YOLANDA FLORES SALGADO

DIRECTOR DEL PROYECTO:

DR. JOSÉ JESÚS CARLOS QUINTANAR SIERRA
FACULTAD DE CIENCIAS, UNAM

CIUDAD UNIVERSITARIA, CD. MX., NOVIEMBRE 2018

A Dios
A mis padres (qepd) ✝
A Armando

Agradecimientos

A la Universidad Nacional Autónoma de México por darme la oportunidad de seguir estudiando en esta grandiosa escuela.

Al Posgrado en Cienca e Ingeniería de la Computación, mis profesores y compañeros de la especialidad.

A mi asesor el Dr. Carlos Quintanar por todas sus enseñanzas, su paciencia y sobre todo su gran apoyo para realizar este proyecto.

A los miembros de la Unidad MOFABI por su apoyo y confianza, muy en especial al Dr. Rubén Ávila, Fernando Valenzuela y Diana Pérez.

A los miembros de mi jurado, el Dr. Luis Miguel De la Cruz y el Mtro. Luciano Díaz por sus observaciones y valiosas recomendaciones.

A la Coordinación de Supercómputo de la DGTIC por las facilidades y apoyo que me dieron para continuar con mis estudios y realizar este proyecto. Las corridas de prueba para la validadión de las aplicaciones OpenFOAM y `DualSPHysics` en comparación con el clúster `gissele` fueron realizadas en la supercomputadora Miztli.

Agradezco al Dr. Armando Rojas Morín y al proyecto PAPIIT IN115416, por el apoyo brindado para la realización de este proyecto.

A Elvia Reyes y Karla Flores por apoyarme en la revisión y correcciones de estilo de este documento.

Índice general

Configuración y puesta en operación de un Clúster HPC para la Unidad MOFABI de la Facultad de Ingeniería de la UNAM

Resumen

El presente proyecto surge a partir de la necesidad que existe, en la Unidad de Modelación de Flujos Ambientales, Biológicos e Industriales (MOFABI), de la Facultad de Ingeniería de la UNAM, de disponer de una plataforma computacional que le permita abordar problemas complejos de manera eficiente y eficaz.

La finalidad de este proyecto es instalar, configurar y poner en operación el equipo de cómputo de alto rendimiento `gissele`, el cual fue adquirido por la unidad MOFABI para realizar simulaciones de mecánica de fluidos y transferencia de calor, así como para instalar las aplicaciones `OpenFOAM` y `DualSPHysics`.

Por lo tanto, se busca que las aplicaciones `OpenFOAM` y `DualSPHysics` se ejecuten con la mayor eficiencia posible.

Introducción

La Computación Científica es un campo multidisciplinario donde se relacionan la modelación de sistemas de ciencia e ingeniería con el uso de computadoras para resolverlos, involucrando tres elementos principales:

- Matemáticas aplicadas a la modelación de los fenómenos del mundo real.
- Análisis numérico: provee los algoritmos para la solución del problema.
- Computación: se encarga de analizar la eficiencia en la implementación de los algoritmos (Eijkhout, 2014).

El Cómputo de Alto Rendimiento (HPC, por sus siglas en inglés) implica la utilización de computadoras para resolver problemas cuya escala (en cantidad de datos o complejidad de operaciones) no hace factible su ejecución en un equipo convencional o de escritorio. El HPC no sólo se relaciona con los algoritmos (códigos) sino con el hardware, donde éstos deben correr de forma eficiente y certera (Hager y Wellein, 2011).

Debido a que los equipos para HPC deben contar con recursos de cómputo cuyo tamaño y complejidad sean superiores a los de un equipo convencional, es necesario y conveniente diseñar, instalar, configurar, mantener y administrar dichos recursos para que sean utilizados en diversas áreas de las ciencias.

La administración de un sistema para HPC implica no sólo el mantenimiento de la infraestructura, sino interactuar con los programadores y científicos para conseguir que las aplicaciones funcionen de manera confiable, eficiente y rápidamente.

Antecedentes

La Unidad de Modelación de Flujos Ambientales, Biológicos e Industriales (MOFABI) es una unidad de investigación que se especializa en la simulación numérica de la mecánica de fluidos, cuya sede se encuentra en el centro de Ingeniería Avanzada de la División de Ingeniería Mecánica e Industrial (DIMEI) de la Facultad de Ingeniería de la UNAM (MOFABI, 2017). Los miembros de la unidad MOFABI realizan proyectos de alcance internacional en mecánica de fluidos, colaborando con investigadores de otras instituciones como la Universidad de California o el Instituto Pakistaní de Ingeniería y Ciencias Aplicadas; incluso, algunos han realizado proyectos, estudios o estancias de investigación tanto en México como en el extranjero, donde han tenido acceso a los equipos HPC. Algunos de ellos han sido usuarios de las supercomputadoras de la UNAM, como la Cray YMP 4/464, la SGI Origin 2000 y la Altix 350.

Con la finalidad de llevar a cabo simulaciones numéricas de fluidos, la unidad MOFABI adquirió en el año 2015 un clúster de supercómputo, el cual actualizó en 2016. Sin embargo, el equipo no había sido puesto en operación debido a la carencia de una adecuada configuración y a la falta de un plan de administración integral del clúster, que permitiera a los usuarios hacer uso del mismo.

El clúster está conformado por los siguientes elementos:

Configuración del clúster		
Nodos / cores Rendimiento	5 Nodos / 100 cores 2.24Tflop/s	HP SL230 G8 con 2 CPU Intel Xeon E5-2680 v2 (2.8 GHz/10 core) 224Gflop/s por CPU / 448Gflop/s por nodo
Memoria RAM	800 GB	160 GB por nodo
Almacenamiento	20 TB	4 discos SAS de 1.2 Terabytes a 10K RPM por nodo
Conexión a Red	10 GbE	2 puertos 10 GbE FLR-T

En su configuración original, cada nodo era un servidor aislado que no compartía recursos con los demás, no contaba con ninguna aplicación instalada ni con una adecuada configuración.

Así, este proyecto surge a partir de la necesidad que tuvo la Unidad MOFABI de poseer un clúster, cuyos recursos puedan ser utilizados en procesos concurrentes y paralelos.

En julio de 2016 se acordó llevar a cabo la configuración del clúster y su puesta en marcha, así como efectuar un curso de capacitación para los miembros de la unidad MOFABI, elaborando un plan de administración del clúster para permitir un adecuado uso del mismo.

Objetivo general

El principal objetivo es configurar y poner en operación el clúster `gissele` de la unidad MOFABI, para realizar simulaciones en la dinámica de fluidos mediante las aplicaciones `OpenFOAM` y `DualSPHysics`.

Para llevar a cabo la validación funcional del clúster, se pretende correr algunas pruebas que muestran el funcionamiento de las aplicaciones `OpenFOAM` y `DualSPHysics`. Y para validar que ambas aplicaciones corran adecuadamente, se comparan los resultados con los datos obtenidos en la supercomputadora Miztli.

Objetivos particulares

Algunos objetivos particulares que se deben tomar en cuenta para un mejor funcionamiento del clúster son:

- Instalar, configurar y poner en funcionamiento el clúster `gissele`.
- Instalar las aplicaciones `OpenFOAM` y `DualSPHysics`.
- Realizar las pruebas de funcionamiento.
- Asesorar a los usuarios en la ejecución de programas en paralelo, utilizando las aplicaciones `OpenFOAM` y `DualSPHysics`.

- Asesorar al grupo de la Unidad MOFABI y a los encargados de cómputo del Centro de Ingeniería Avanzada sobre los procedimientos básicos y la administración del clúster `gissele`, con el objetivo de que lo mantengan en operación.

Alcances

El proyecto consiste en poner en funcionamiento el clúster con una distribución GNU/Linux `CentOS`, con las aplicaciones `OpenFOAM` y `DualSPHysics` instaladas.

Así mismo, la solución propuesta para el almacenamiento será la que combine mejor la confiabilidad y el desempeño, utilizando los recursos y la configuración de hardware disponibles.

Limitaciones

En el caso de la aplicación `DualSPHysics` y debido a que el clúster no cuenta con tarjetas GPU, sólo se configurará para funcionar en la modalidad de procesamiento en CPU.

Es conveniente proponer una configuración que permita utilizar el clúster en la mejor forma posible, empleando para ello los recursos disponibles con que se cuenta y las herramientas *Open Source*.

Metodología

Para llevar a cabo el proyecto, se desarrollaron los siguientes temas:

1. Análisis de la situación inicial
2. Propuesta de la configuración del clúster
3. Instalación y configuración del clúster
4. Instalación de las aplicaciones OpenFOAM y DualSPHysics
5. Ejecución de pruebas de las aplicaciones
6. Comparación de los resultados obtenidos
7. Relación de las asesorías para el uso y mantenimiento del clúster
8. Propuestas futuras para el mejoramiento de la infraestructura

1. Análisis de la situación inicial

El clúster `gissele`, de la unidad MOFABI, está compuesto por cinco nodos de procesamiento que reúnen las siguientes características:

Configuración de un nodo	
Modelo y procesador	HP SL230 G8 con 2 CPU Intel Xeon E5-2680 v2 (2.8 GHz/10 core)
Memoria RAM	160 GB
Almacenamiento	4 discos SAS de 1.2 Terabytes a 10K RPM
Conexión a Red	2 puertos 10 GbE FLR-T

Al iniciar el proyecto, cada nodo tenía la siguiente configuración:

Sistema	
Sistema Operativo	Red Hat Enterprise Linux ComputeNode release 7.1 (Maipo)
Espacio de almacenamiento	
1er. Disco Duro	/boot 2.2G
	/ 54G
	/home 43G
	/var 22G
	Resto del disco en un volumen lógico sin usar
2o. Disco Duro	Volumen lógico sin usar
3er. Disco Duro	Volumen lógico sin usar
4o. Disco Duro	Volumen lógico sin usar
Red	
1er. Puerto 10GbE	Con entrada/salida a la red interna del CIA
2o. Puerto 10GbE	Sin configurar

Problemas detectados en el clúster:

1. Nodos desactualizados, sin la posibilidad de su actualización debido a la expiración de la licencia de Red Hat Enterprise Linux (RHEL).
2. Nodos independientes que no estaban configurados para compartir archivos:
   a) Cada nodo tenía su propio directorio HOME.
   b) Como los nodos son independientes e iguales, no había roles asignados entre ellos.
3. Cada uno de los nodos estaba conectado al *switch* principal del edificio del Centro de Ingeniería Avanzada, en la red interna del mismo, pero sin que existiera un segmento propio de la red, por lo que cada uno era visible desde cualquier máquina de la unidad MOFABI.
4. El clúster se encuentra aislado de Internet por las reglas del *firewall* principal de la red del edificio, por lo que no existe salida ni entrada hacia/desde Internet. Todas las direcciones IP son privadas.
5. El clúster no contaba con herramientas de seguridad o administración.

Limitaciones encontradas al efectuar el análisis:

Al realizar el análisis de la problemática, se encontraron las siguientes limitaciones que no podrían ser resueltas debido a diversos factores, siendo el principal el de tipo presupuestal.

1. No se cuenta con licencias de software ni el presupuesto para adquirirlas, lo que llevó a la determinación de solucionar el problema empleando un software libre.
2. La Unidad MOFABI adquirió el clúster, pero no compró el equipo de red necesario para configurarlo. El *switch* 10GbE pertenece al área de cómputo del Centro de Ingeniería Avanzada y es fundamental para los usuarios del edificio. Actualmente no es posible contar con un *switch* que pudiera dedicarse exclusivamente al clúster.
3. Debido a que el *switch* no pertenece a la Unidad MOFABI, no es posible reconfigurarlo para tener un segmento dedicado en su totalidad al clúster.

2. Propuesta de la configuración del clúster

Para la propuesta de configuración y poner a punto el clúster, se tomaron en cuenta las limitaciones arrojadas por el análisis preliminar. Con base en ello, se determinó lo siguiente:

Configuración General del Sistema

En la propuesta de configuración se tomó como base el esquema general de un clúster, el cual a su vez está basado en un modelo cliente/servidor.

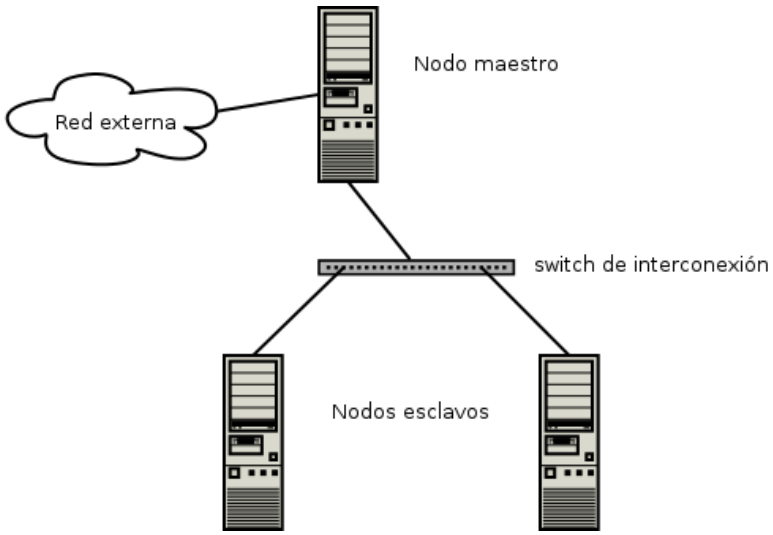


Figura 1. *Esquema general de un clúster.*

Roles

El nombre asignado al clúster será el de: `gissele`.

Cada nodo físico será nombrado (hostname) como: `gissele_1`, `gissele_2`, `gissele_3`, `gissele_4` y `gissele_5`, respectivamente.

Por consiguiente, a cada nodo se le asignará un rol específico dentro del clúster:

Nodo de login: Se denomina así a la interfaz que permite la comunicación entre el clúster y la red externa, la cual controla el acceso y la validación de los usuarios.

Nodo maestro: Se encarga de proporcionar servicios a los nodos de cómputo, así como realizar las funciones de monitoreo y aprovisionamiento de dichos nodos, lo cual permite a los usuarios enviar sus trabajos a los nodos de cómputo.

Nodo de cómputo: Se conoce también como *nodo esclavo*, y se utiliza con el fin de realizar cálculos numéricos.

De acuerdo con el rol que desempeñe cada nodo, serán conocidos (alias) como `gis[1-6]`, donde el nodo `gissele_1` se denominará `gis1` o `gis6`, respectivamente, dependiendo del rol que asuma cada uno.

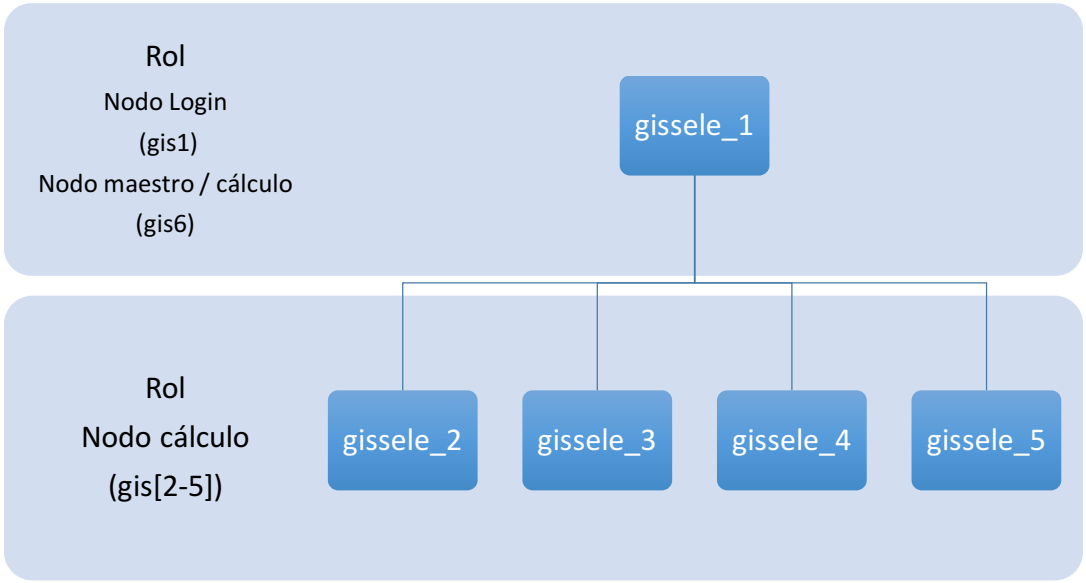


Figura 2. *Roles asignados a los diferentes nodos.*

Los servicios proporcionados por el nodo `gissele_1` serán los siguientes:

Nodo	**Rol**	**Servicios**
gissele_1	Nodo de *login* (gis1) Nodo maestro (gis6)	Acceso a los usuarios a través de `ssh` Proporciona los servicios de: • NTP • NFS / Sistemas de archivos • Repositorio de software • Administración de tareas

Sistema Operativo

Para efectuar el análisis se utilizó como sistema operativo el `CentOS Linux release 7.3.1611`, cuya versión es la más actual del sistema `CentOS`. La decisión se tomó con el fin de tener un sistema operativo basado en `RHEL` actualizado, que permitiera una mayor estabilidad y seguridad.

Espacio de almacenamiento

Cada uno de los nodos cuenta con cuatro discos duros de 1TB.

El espacio de almacenamiento fue repartido de la siguiente forma:

Espacio de almacenamiento	
1er. Disco Duro	/boot 512M ext4 / 100G ext4 swap 32G swap /home 1068G ext 4 **solo gissele_1* /respaldos 1068G ext 4 **nodos gissele_[2-5]*
2o. Disco Duro	Una sola partición sin punto de montaje
3er. Disco Duro	Una sola partición sin punto de montaje
4o. Disco Duro	Una sola partición sin punto de montaje

Red

Debido a las limitantes de no contar con un *switch* propio, se tomó la determinación de aislar lo más posible el clúster.

En el caso del nodo `gissele_1` (*login*, maestro), se reconfiguró de la siguiente forma:

Red	
1er. Puerto 10GbE Rol: maestro	Con entrada/salida a redUNAM gissele_1 (gis1)
2o. Puerto 10GbE Rol. *login*	Con entrada/salida a la red interna del CIA gissele_1 (gis6)

En lo referente a los nodos `gissele_[2-5],` únicamente se activó el 2º. Puerto 10GbE con entrada/salida a la red interna del CIA.

Configuración de los sistemas de archivos

Sistema de archivos HOGAR (HOME)

El sistema de archivos `/home` es un conjunto ordenado de archivos donde se encuentran los directorios HOGAR de cada usuario.

El nodo `gissele_1` (gis6) comparte este sistema de archivos, por medio del servicio `NFS`. Su capacidad utilizable es de 979 G.

Sistema de archivos Datos

Cada uno de los nodos cuenta con tres discos de 1TB.

Debido a que no se cuenta con un sistema o nodo específico para el almacenamiento centralizado, se utilizaron los discos duros restantes de cada nodo para destinarlos a efectuar dicho almacenamiento.

Redundancia

Los nodos cuentan con el auxilio de controladoras para los arreglos redundantes de discos independientes o (RAID por sus siglas en inglés) tipo *HP Dynamic Smart Array B320i controller*; sin embargo, no fue posible activarlas debido a que los nodos no cuentan con los cables conectores apropiados para los discos, los cuales se venden en forma opcional y no fueron adquiridos al comprar los nodos.

Se pensó en la posibilidad de construir un RAID mediante un software, para lo cual se estudiaron diversas alternativas tales como `mdadm` (*Multiple devices admin*) o `zfs` con `raidz`; incluso, se realizaron pruebas de configuración pero se llegó a la conclusión de que, debido a lo reducido del espacio en el disco, sería conveniente sugerir la adquisición de un nuevo nodo para el almacenamiento exclusivamente, en un futuro cercano.

Como la construcción de un sistema RAID por software implicaba reducir aún más el espacio de almacenamiento con que cuenta el clúster `gissele`, se tomó la decisión de sacrificar la redundancia para tener disponible mayor espacio en el disco.

Tipos de sistema de archivos considerados

Para el efecto, fueron revisadas dos propuestas de sistema de archivos:

GFS2

El sistema de archivos GFS2 (*Global File System 2*) es un sistema de archivos distribuido, cuya característica principal es que cada nodo `GFS2` comparte dispositivos de almacenamiento en bloques, los cuales servirán para integrar un sistema único de archivos en forma global.

Dicho modelo está diseñado para sistemas tipo SAN (*Storage Area Network*) o servidores con hardware de alta calidad, aunque es factible correrlo en cualquier tipo de máquina. La capacidad máxima de un sistema de archivos `GFS2` es de 100 TB y solamente se recomienda para clústeres de 2 a 16 nodos (Red Hat, 2009; Levine, 2013).

En la actualidad, el sistema de archivos `GFS2` es mantenido y desarrollado por Red Hat; por ello, es compatible con el sistema `CentOS7`.

Gluster

El sistema de archivos `Gluster` es multiescalable, ya que permite agregar varios servidores de sistemas de archivos en un entorno de archivos en paralelo.

En realidad, los sistemas de archivos (llamados *bricks*) son sistemas de archivos ya existentes en los nodos servidores y se comparten por medio de la red en un espacio único de sistemas de archivos.

`Gluster` suele utilizar el "espacio de usuario", y debido a ello se originan ligeras sobrecargas en los cambios de contexto en archivos pequeños, pero trabaja muy bien con los archivos grandes. Asimismo, su funcionamiento se da en un ambiente cliente/servidor. Por defecto, los archivos se almacenan enteros, aunque existen opciones para distribuirlos (*stripping*).

El sistema `Gluster` es un software libre, propiedad de Red Hat, pero es mantenido por los usuarios; por lo tanto, es compatible con `CentOS7` (Anoop, C. S., 2014; Porkolab y Bear, s.f.).

Debido a que el sistema de archivos sería construido sin el apoyo de una tecnología de redundancia, se tomó la decisión de utilizar `Gluster`, ya que ante la pérdida de un disco duro se minimizarían los daños ocasionados a la información, debido a que los sistemas de archivos son independientes y se puede acceder a ellos por medio de cada uno de los nodos.

Así, la configuración propuesta para los nodos en cuanto al espacio de almacenamiento obtuvo los siguientes resultados:

Espacio de almacenamiento	
1er. Disco Duro	/boot 512M ext4 / 100G ext4 swap 32G swap /home 1068G ext 4 *solo gissele_1 /respaldos 1068G ext 4 *nodos gissele_[2-5]
2o. Disco Duro	/brick1
3er. Disco Duro	/brick2
4o. Disco Duro	/brick3

Administración de tareas

Cuando se toma la decisión de trabajar en un clúster (en un ambiente distribuido), para alcanzar un alto desempeño se requiere de un mecanismo con mayor eficiencia (más allá del sistema operativo), que permita una asignación dinámica de los recursos, cuya finalidad es distribuir la carga de acuerdo con el estado del sistema. A la herramienta que permite realizar el balance de carga se le conoce como "Sistema de Administración de Tareas" (*Resource Management System/ Job Management System*), (Flores y García, 2003).

En un sistema ideal, cada procesador debería ejecutar la misma cantidad de trabajo, teniendo la certeza de que todos los procesadores sean utilizados. El balanceo de carga permite mejorar el uso de los recursos y, por tanto, el desempeño de un sistema.

Con el fin de mejorar el uso del clúster `gissele`, se hizo la propuesta de instalar un Sistema de Administración de Tareas. Para tal efecto, fueron evaluados los siguientes:

Torque (Terascale Open-source Resource and QUEue Manager)

Es un Sistema de Administración de Tareas creado en 2003, el cual está basado en PBS (*Portable Batch System*) y su uso ha sido ampliamente probado en sistemas HPC. En la actualidad, dicho sistema es mantenido por Adaptative Computing y se considera casi un estándar para administradores de tareas; sin embargo, durante la realización de este proyecto notamos que Torque ya no se distribuye libremente, por lo que se decidió no utilizarlo.

Slurm (Simple Linux Utility for Resource Management)

Slurm es un sistema de programación de tareas y de gestión de clústeres de código abierto; posee gran tolerancia a las fallas y es altamente escalable para clústeres grandes y pequeños de Linux (SchedMD, 2013).

Debido a que uno de los requerimientos iniciales era que la propuesta debía estar basada en un software libre, se tomó la decisión de instalar el sistema `slurm`.

3. Instalación y configuración del clúster

A continuación se describe el proceso de configuración, derivado de la instalación del clúster `gissele`. Para efectuar dicha instalación, se tomaron en cuenta las recomendaciones de configuración propuestas y algunas sugerencias hechas para el diseño y la administración de clústeres por parte del Laboratorio de Clusters y Grids (Díaz, 2016).

Sistema operativo

Sistema operativo	CentOS Linux
Versión	7.3.1611
Arquitectura	64 bits
Media de instalación	CentOS-7-x86_64-DVD-1611.iso
Sha256sum	c455ee948e872ad2194bdddd39045b83634e8613249182b88f549bb2319d97eb

Red

En el siguiente esquema se muestra la configuración definida para efectuar la conexión a red del clúster `gissele`:

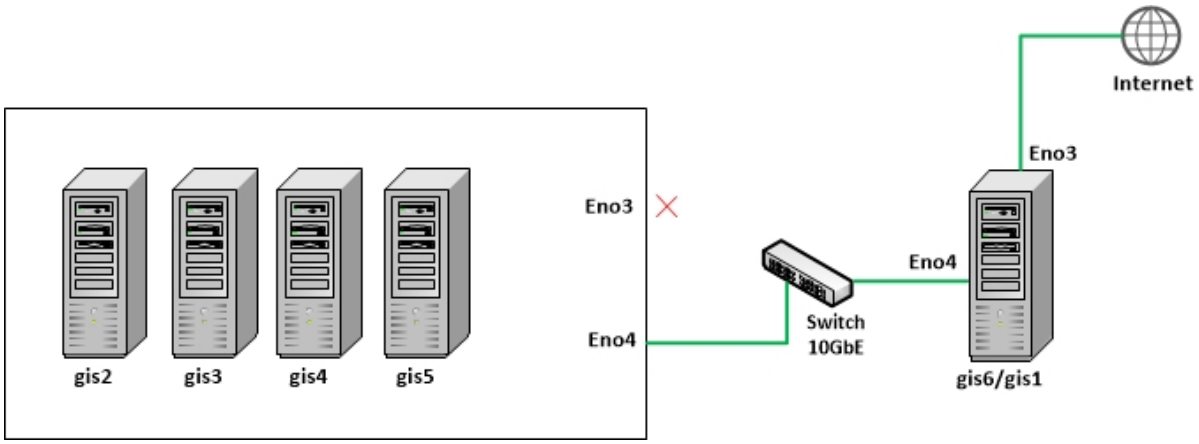


Figura 3. *Esquema general de conexión del clúster* gissele.

Interfaces de red

En seguida se muestran las interfaces de red de los diferentes nodos, el modelo y el estado en que se encuentran dichas interfaces.

Nodo `gissele_1`

Interfaz	Modelo	Estado
eno3	Broadcom corporation NetXtreme II BCM57810 10 Gigabit Ethernet (gis1)	Activa
eno4	Broadcom corporation NetXtreme II BCM57810 10 Gigabit Ethernet (gis6)	Activa

Nodos `gissele_[2-5]`

Interfaz	Modelo	Estado
eno3	Broadcom corporation NetXtreme II BCM57810 10 Gigabit Ethernet	Inactiva
eno4	Broadcom corporation NetXtreme II BCM57810 10 Gigabit Ethernet (gis[2-5])	Activa

Configuración

Host	eno3	eno4	Gateway	Servidor DNS
gissele_1 gissele_6	172.16.14.1/24	172.16.14.6/24	172.16.14.254	132.248.10.2
gissele_2		172.16.14.2/24	172.16.14.254	132.248.10.2
gissele_3		172.16.14.3/24	172.16.14.254	132.248.10.2
gissele_4		172.16.14.4/24	172.16.14.254	132.248.10.2
gissele_5		172.16.14.5/24	172.16.14.254	132.248.10.2

* Para cada uno de los nodos se desactivó y eliminó el servicio NetworkManager.

Configuraciones y servicios básicos

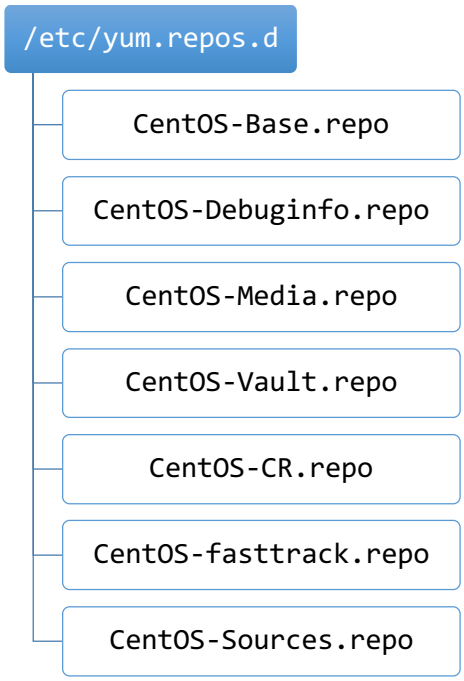
Configuraciones en el nodo maestro/*login* `gissele_1`

Paso 1. Definición de los repositorios

Los repositorios son los lugares/contenedores donde se encuentran los paquetes y programas, para una distribución de Linux.

Algunos ya vienen previamente definidos por la distribución; en este caso, al instalar CentOS 7 por omisión, se tendrán los repositorios que están contenidos en servidores en la Internet, a saber:

Repositorios CentOS



Repositorios EPEL (Paquetes para Linux Empresarial)

Actualmente existe un repositorio denominado EPEL, el cual contiene algunos paquetes adicionales de alta calidad para Linux Empresarial; éstos son mantenidos por un grupo de interés especial de Fedora y son compatibles con las distribuciones Red Hat Enterprise Linux, CentOS y Scientific Linux.

Algunas herramientas que facilitan la administración de Linux están contenidas en estos paquetes, por lo que se recomienda su instalación:

Repositorios locales

Como los nodos de cómputo `gissele_[2-5]` no tienen salida a Internet, no pueden acceder a los repositorios anteriores; por lo tanto, con el fin de facilitar la instalación del software en los mismos, se tomó la decisión de crear algunos repositorios locales.

La configuración de éstos repositorios se realizó en el nodo maestro `gissele_1`, para ello se crearon los siguientes directorios:

`/home/repositorio/Centos7-local`

Proporciona los paquetes generales de CentOS 7 contenidos en el medio de instalación original CentOS-7-x86_64-DVD-1611.iso.

`/home/repositorio/epel-local`

Se encarga de proporcionar los paquetes generales EPEL que se instalen en el clúster.

`/home/repositorio/Otros-local`

Proporciona cualquier paquete de software obtenido de otros repositorios diferentes a los repositorios base de CentOS o EPEL.

Para cada directorio se ejecutó el siguiente comando:

```
createrepo .
```

Paso 2. Desactivación de SELinux

- Se modificó la configuración para desactivar el módulo de seguridad integrado SELinux. Para ello, se cambió el valor de la variable SELINUX en el archivo de configuración `/etc/sysconfig/selinux.`

  `SELINUX=disabled`

Paso 3. Configuración de servicios

iptables

- Se desactivó el servicio `firewalld` y se instaló el servicio `iptables`

Este cambio se hizo así para facilitar la administración, debido a que el personal del CIA requiere el acceso al equipo para modificar este tipo de reglas. Como `firewalld` es una herramienta que solamente está disponible por omisión en sistemas basados en RedHat 7 o superior, se decidió utilizar el servicio `iptables` en la configuración *firewall* (en lugar de `firewalld`), debido a su gran compatibilidad en la administración con otros servidores Linux.

Para desactivar `firewalld` se procedió a realizar los siguientes pasos:

```
systemctl stop firewalld
systemctl mask firewalld
systemctl disable firewalld
```

Una vez desactivado `firewalld` se instaló el servicio `iptables`:

```
yum install iptables-services
systemctl enable iptables
systemctl start iptables
```

El archivo que se configuró fue el siguiente:

`/etc/sysconfig/iptables`

Para la interfaz eno3 sólo se permite el acceso al puerto 22. El filtrado de las direcciones externas se hace a través del *firewall* general de la dependencia:

```
-A INPUT -i eno3 -p tcp -m state --state NEW -m tcp --dport 22 -j
ACCEPT
```

Solamente se permiten los accesos desde las IPs de los nodos gis[1-6]:

```
-A INPUT -s 172.16.14.1 -m state --state NEW -j ACCEPT
-A INPUT -s 172.16.14.2 -m state --state NEW -j ACCEPT
-A INPUT -s 172.16.14.3 -m state --state NEW -j ACCEPT
-A INPUT -s 172.16.14.4 -m state --state NEW -j ACCEPT
-A INPUT -s 172.16.14.5 -m state --state NEW -j ACCEPT
-A INPUT -s 172.16.14.6 -m state --state NEW -j ACCEPT
```

Debido a que el *switch* es compartido, fue necesario restringir el acceso por `ssh` a cualquier otro equipo de la red interna, y sólo se activó para ciertas direcciones IP. Para ello, utilizamos los siguientes archivos:

`/etc/hosts.deny`

```
ALL:ALL
```

`/etc/hosts.allow`

```
ALL:127.0.0.1,172.16.14.1,172.16.14.2,172.16.14.3,172.16.14.4,172
.16.14.5,172.16.14.6
sshd:132.248.X.X
```

NTP (Network Time Protocol)

El NTP es un protocolo diseñado para sincronizar el tiempo en las computadoras que conforman una red. Los equipos de la red consiguen el tiempo de una fuente válida, tal como un radio, un reloj o un reloj atómico asociado a un servidor de tiempo (servidor NTP).

El servidor NTP distribuye su tiempo a través de la red. Un cliente NTP realiza una transacción con su servidor, sobre un intervalo de sondeo (a partir de 64 a 1024 segundos) que cambie de manera dinámica en determinado plazo, dependiendo de los estados de la red entre el servidor NTP y el cliente (Cisco, 2008).

En este caso, el nodo gissele_1 será configurado para ambos casos.

Un cliente, al sincronizar su reloj con los servidores `cronos.cenam.mx` y `1.centos.pool.ntp.org` y un servidor distribuyendo su tiempo a los nodos `gissele_[2-5]`:

`/etc/ntp.conf`

```
restrict 172.16.14.0 mask 255.255.255.0 nomodify notrap
server cronos.cenam.mx iburst
server 1.centos.pool.ntp.org iburst
```

Al activar el servicio muestra lo siguiente:

```
systemctl enable ntpd
systemctl start ntpd
ntpdate cronos.cenam.mx
```

Servicio Pdsh

El servidor *Parallel Distributed Shell* es un cliente Shell remoto multihebra, el cual permite ejecutar comandos en múltiples *hosts* remotos en paralelo.

Primero se dieron de alta los nodos en el archivo `/etc/hosts`, como *hosts* conocidos:

```
172.16.14.1     gissele_1       gis1
172.16.14.2     gissele_2       gis2
172.16.14.3     gissele_3       gis3
172.16.14.4     gissele_4       gis4
172.16.14.5     gissele_5       gis5
172.16.14.6     gissele_6       gis6
```

Después se configuró el archivo `/etc/pdsh/nodos` de la siguiente manera:

```
gis1
gis2
gis3
gis4
gis5
gis6
```

`/etc/profile.d/pdsh.sh`

```
export PDSH_RCMD_TYPE='ssh'
export WCOLL='/etc/pdsh/nodos'
```

Se generó la llave pública y se envió una copia al archivo `authorized_keys`

```
ssh-keygen -t rsa
ssh-copy-id gis1
```

Se recolectaron la llaves de identificación de los nodos y se puso el resultado en `/etc/ssh/ssh_known_hosts`

```
for i in $(seq 6); do ssh-keyscan -t rsa gis$i,172.16.14.$i >>
/etc/ssh/ssh_known_hosts; done
```

NFS

Network File System: el Sistema de Archivos en Red permite a los *hosts* remotos montar sistemas de archivos sobre la red e interactuar con ellos como si estuvieran montados localmente. Esto permite la consolidación de los recursos en los servidores centralizados en la red (Red Hat, 2005).

Para configurar el servicio NFS en el nodo `gissele_1`, se incluyeron los paquetes NFS (`nfs-utils.x86_64 y dependencias`) desde la instalación, por lo que solamente fue verificado el estado del servicio:

```
systemctl status nfs
```

Se exportó el sistema de archivos `/home` a los nodos restantes.

La edición del archivo de configuración `/etc/exports`, queda de la siguiente forma:

```
/home 172.16.14.2(sync,rw,no_root_squash)
/home 172.16.14.3(sync,rw,no_root_squash)
/home 172.16.14.4(sync,rw,no_root_squash)
/home 172.16.14.5(sync,rw,no_root_squash)
```

En seguida se debe activar:

```
exportfs -a
```

Otras utilerías relevantes

Se instalaron los siguientes programas, los cuales son de gran utilidad para la administración del clúster:

- **lshw:** permite obtener un listado detallado del hardware del sistema.

- **screen:** Permite multiplexar las terminales; es muy útil para la administración remota, pues la terminal se ejecuta en un segundo plano, de tal forma que aunque se pierda la conexión la terminal sigue activa. Incluso, en cualquier momento los procesos que alguna terminal está ejecutando, sin que éstos se pierdan.

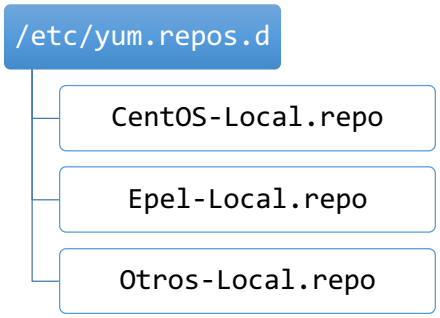
- **environment-modules:** Permite modificar en forma dinámica el entorno de un usuario. Es de gran utilidad para proveer de un entorno adecuado de trabajo a los usuarios, por medio de módulos que se pueden cargar o descargar.

Configuraciones en los nodos de cómputo `gissele_[2-5]`

Paso 1. Configuración de los repositorios

Repositorios locales

Se configuraron los repositorios locales creados en el nodo maestro, debido a que los nodos de cómputo no tienen acceso a Internet.



Paso 2. Desactivación de SELinux

- Se modificó la configuración para desactivar el módulo de seguridad integrado `SELinux`. Para ello, se cambió el valor de la variable SELINUX en el archivo de configuración `/etc/sysconfig/selinux.`

  ```
  SELINUX=disabled
  ```

Paso 3. Configuración de servicios

Iptables

Al igual que en el nodo maestro, se desactivó el servicio `firewalld` y se instaló `iptables`.

Se utilizaron las mismas reglas de `iptables` que en el nodo maestro (véase el Paso 3. Configuración de servicios para el nodo `gissele_1`).

NTP (Network Time Protocol)

Los nodos `gissele_[2-5]` fueron configurados para sincronizar sus relojes, a partir del nodo `gissele_1` (`gis6`). Para ello, se efectuó el siguiente paso:

Modificación del archivo:

`/etc/ntp.conf`

```
restrict 172.16.14.0 mask 255.255.255.0 nomodify notrap
server gis6 iburst
```

Activación del servicio:

```
systemctl enable ntpd
systemctl start ntpd
ntpdate gis6
```

Ssh

Para ayudar en la identificación de los nodos del clúster, se copió el archivo `/etc/ssh/ssh_known_hosts` del nodo gissele_1 a los nodos `gissele_[2-5]`.

NFS

Cada uno de los nodos cliente debe montar el sistema de archivos `/home`.

Fue modificada la tabla `/etc/fstab`, añadiendo la siguiente línea:

```
172.16.14.6:/home     /home nfs4  rw,hard,intr 0 0
```

Se montó el sistema de archivos `/home`:

```
mount /home
```

Otras utilerías relevantes

Se instalaron los siguientes programas de gran utilidad:

- lshw
- environment-modules

Configuración del sistema de archivos GlusterFS

La configuración del sistema de archivos gluster `/datos`, se realizó de acuerdo con las siguientes características:

a) Los *bricks* o directorios a ser compartidos en el fondo de almacenamiento confiable (*trusted storage pool*) serán los directorios `brick`, en los sistemas de archivos que fueron creados a partir de los tres discos adicionales de cada nodo y se montaron localmente en los puntos `/brick1`, `/brick2` y `/brick3`, respectivamente.
b) Un volumen es una colección lógica de *bricks*. Para este caso, el tipo de volumen seleccionado es un "volumen distribuido". En dicho volumen los archivos se almacenan completos, pero el sistema intenta balancear la carga en los diferentes nodos que componen el fondo de almacenamiento.
c) No se incluyó la redundancia para no sacrificar más espacio en el disco, como se mencionó en la propuesta de la configuración, por lo que no existe réplica de datos.

Instalación de Gluster

La instalación del sistema de archivos Gluster se hizo a partir de los paquetes *CentOS Storage SIG*.

Para el nodo `gissele_1` queda lo siguiente:

```
sudo yum install centos-release-gluster310
sudo yum install glusterfs gluster-cli glusterfs-libs glusterfs-server
```

En cada uno de los nodos `gissele_[2-5]` se aprecia lo siguiente:

```
sudo yum install glusterfs gluster-cli glusterfs-libs glusterfs-server
```

Activación del servicio Gluster

```
sudo systemctl enable glusterd
sudo systemctl start glusterd
```

Configuración del fondo confiable de almacenamiento Gluster Trusted Storage Pool

Para cada uno de los nodos `GlusterFS`, se efectúa la conexión en el nodo `gissele_1`:

```
sudo gluster peer probe gis2
sudo gluster peer probe gis3
sudo gluster peer probe gis4
sudo gluster peer probe gis5
sudo gluster peer probe gis6
```

Se lleva a cabo una prueba de la conexión:

```
sudo gluster peer status
```

De donde resulta una salida parecida a la siguiente:

```
Number of Peers: 4

Hostname: gis2
Uuid: f4d95805-6276-4b52-b542-3d38bea147af
State: Peer in Cluster (Connected)
```

Creación del volumen distribuido Distributed GlusterFS Volume

El nombre asignado al volumen será: `glv0`

Los siguientes comandos solamente se ejecutan en el nodo `gissele_1`:

```
sudo gluster volume create glv0 gis6:/brick1/brick
gis6:/brick2/brick gis6:/brick3/brick

sudo gluster volume start glv0
```

Después se añaden los *bricks* de cada uno de los nodos con las instrucciones siguientes:

```
sudo gluster volume add-brick glv0 gis3:/brick1/brick
sudo gluster volume add-brick glv0 gis3:/brick2/brick
sudo gluster volume add-brick glv0 gis3:/brick3/brick
```

Montaje del sistema de archivos

Para cada nodo `gis[2-6]`

Se modifica el archivo `/etc/fstab`, añadiendo la siguiente línea:

```
gis6:/glv0 /datos glusterfs defaults,direct-io-
mode=disable,_netdev,backupvolfile-server=gis2 0 0
```

Posteriormente se monta el sistema `/datos`:

```
sudo mount -t glusterfs gis6:/glv0 /datos
```

En seguida se verifica el sistema de archivos:

```
sudo gluster volume info
```

Lo cual producirá una salida como la siguiente:

```
Volume Name: glv0
Type: Distribute
Volume ID: f8848d83-ec34-43a2-a47c-cc7a38c8bdb8
Status: Started
Snapshot Count: 0
Number of Bricks: 15
Transport-type: tcp
Bricks:
Brick1: gis3:/brick1/brick
Brick2: gis3:/brick2/brick
Brick3: gis3:/brick3/brick
Brick4: gis6:/brick1/brick
Brick5: gis6:/brick2/brick
Brick6: gis6:/brick3/brick
```

Reconfiguración

Con el fin de mejorar el rendimiento del sistema, se llevaron a cabo las siguientes reconfiguraciones:

```
sudo gluster volume set glv0 client.event-threads 10
sudo gluster volume set glv0 server.event-threads 10
sudo gluster volume set glv0 performance.cache-size 1GB
sudo gluster volume set glv0 performance.cache-max-file-size 128MB
sudo gluster volume set glv0 performance.readdir-ahead on
sudo gluster volume set glv0 performance.parallel-readdir on
```

Configuración del sistema de administración de tareas slurm

La configuración de `slurm` se realizó de forma muy básica, con el fin de mejorar la administración de los recursos y dar un mayor seguimiento a los requerimientos de los usuarios.

Instalación slurm

Creación de los usuarios (`gissele_[1-5]`):

```
export MUNGEUSER=991
groupadd -g $MUNGEUSER munge
useradd  -m -c "MUNGE" -d /var/lib/munge -u $MUNGEUSER -g munge  -s /sbin/nologin
munge
export SLURMUSER=990
groupadd -g $SLURMUSER slurm
```

```
useradd  -m -c "SLURM workload manager" -d /var/lib/slurm -u $SLURMUSER -g slurm -s /bin/bash slurm
```

Instalación de `munge` para autenticación (`gissele_[1-5]`):

```
yum install munge munge-libs munge-devel rng-tools
```

Creación de la llave (`gissele_1`):

```
rngd -r /dev/urandom
/usr/sbin/create-munge-key –r
dd if=/dev/urandom bs=1 count=1024 > /etc/munge/munge.key
chown munge: /etc/munge/munge.key
chmod 400 /etc/munge/munge.key
```

Se copió la llave a los nodos de cómputo (`gissele_[2-5]`):

```
pdsh –w gis[2-5] /etc/munge/munge.key  /etc/munge/munge.key
```

Para iniciar el servicio `munge` en cada uno de los nodos, primero se prepararon los directorios:

```
chown -R munge: /etc/munge/ /var/log/munge
chmod 0700 /etc/munge/ /var/log/munge
```

```
systemctl enable munge
systemctl start munge
```

Instalación de los pre-requisitos

Este paso se refiere a la instalación de las dependencias que están especificadas en el manual de instalación (SchedMD, 2018):

```
yum install hwloc hwloc-devel libibmad libibumad man2htm ncurses-devel numactl numactl-devel openssl openssl-devel pam-devel readline-devel rrdtool-devel mariadb mariadb-devel rpm-build
```

Construcción de rpms

El software de `slurm` se obtuvo con el siguiente comando, cuya versión 17.11.9 era la última en ese momento:

```
wget http://www.schedmd.com/download/latest/slurm-17.11.9-2.tar.bz2
```

El mecanismo de instalación empleado fue a partir de la construcción de los *rpms* de `slurm`, para ello se ejecutó el siguiente comando:

```
rpmbuild -ta slurm-17.11.9-2.tar.bz2
```

Los *rpms* resultantes fueron copiados al repositorio Otros-Local.

Instalación de paquetes de slurm

En el nodo maestro (gissele_1):

```
slurm slurm-example-configs slurm-pam_slurm slurm-perlapi slurm-
slurmctld slurm-slurmdbd
```

En los nodos de cómputo (gissele_[2-6]):

```
slurm slurm-perlapi slurm-slurmd
```

Configuración de slurm

Para configurar `slurm` de una forma rápida, se construyó la primera versión del archivo /etc/slurm/slurm.conf con la ayuda de la herramienta configurator_easy.

Dicha herramienta se corre desde un navegador en cualquier PC, conectándose a https://slurm.schedmd.com/configurator.easy.html, permitiendo generar el contenido del archivo de configuración.

Una vez construido el archivo de forma básica, se fue modificando conforme al manual y a la sección de configuración de `slurm`; finalmente quedó de la siguiente forma:

```
AuthType=auth/munge
ClusterName=mofabi
ControlMachine=gissele_6
SlurmUser=slurm
SwitchType=switch/none
#
StateSaveLocation=/var/spool/slurm/ctld
#
SlurmctldPort=6817
SlurmctldPidFile=/var/run/slurmctld.pid
#
SlurmdPort=6818
SlurmdPidFile=/var/run/slurmd.pid
SlurmdSpoolDir=/var/spool/slurm/d
#
```

```
MpiDefault=none
ProctrackType=proctrack/linuxproc
#ProctrackType=proctrack/cgroup
#ReturnToService=1
#SlurmdUser=root
TaskPlugin=task/affinity
#
# TIMERS
KillWait=30
SlurmctldTimeout=300
SlurmdTimeout=300
#
#
# SCHEDULING
FastSchedule=1
SchedulerType=sched/backfill
SelectType=select/cons_res
SelectTypeParameters=CR_Core
#
#
# LOGGING AND ACCOUNTING
SlurmctldDebug=3
SlurmctldLogFile=/var/spool/slurm/log/slurmctld.log
SlurmdDebug=3
SlurmdLogFile=/var/spool/slurm/log/slurmd.log
AccountingStorageType=accounting_storage/filetxt
JobAcctGatherFrequency=30
JobAcctGatherType=jobacct_gather/linux
#
#
# COMPUTE NODES
NodeName=gissele_[2-6] RealMemory=96479 Sockets=2
CoresPerSocket=10 ThreadsPerCore=1
PartitionName=cola Nodes=gissele_[2-6] Default=YES
MaxTime=INFINITE State=UP
```

Posteriormente, este archivo resultante se copia a los nodos de cómputo.

Se debe verificar que el archivo, los directorios y archivos mencionados en él, así como el archivo `/var/log/slurm_jobacct.log` existan y pertenezcan al usuario `slurm`.

Iniciar los servicios de slurm

En el nodo maestro:

```
systemctl enable slurmctld.service
systemctl start slurmctld.service
systemctl status slurmctld.service
```

En los nodos de cómputo:

```
systemctl enable slurmd.service
systemctl start slurmd.service
systemctl status slurmd.service
```

4. Instalación de las aplicaciones OpenFOAM y DualSPHysics

Debido a que la Unidad MOFABI se especializa en temas de mecánica de fluidos y siendo la dinámica de fluidos una subrama de esta especialidad, fue necesario instalar dos aplicaciones que permitieran abordar diversos problemas en esta área, obteniendo simulaciones numéricas y visualizaciones tridimensionales. Aunque la visualización no se realiza propiamente en el clúster, sí se procesan y se generan los archivos necesarios para visualizarlos.

Las aplicaciones requeridas para su instalación fueron: `OpenFOAM` y `DualSPHysics`.

OpenFOAM

Esta aplicación es de tipo de software libre y sirve para la simulación en la dinámica de fluidos computacional (CFD), es desarrollada por la *OpenFOAM Fundation* y distribuida bajo la licencia GPL (*GNU Public License*) (CFD Direct, s.f.). Está desarrollada principalmente en C++.

Los solucionadores (*solvers*) de `OpenFOAM` se clasifican por tipo de fluidos, en los siguientes:

- Flujos incompresibles
- Flujos multifásicos
- Combustión
- Flujos gobernados por flotación
- Transferencia de calor
- Flujos compresibles
- Métodos de partícula

Para instalar `OpenFOAM` se decidió la compilación desde el código fuente, sin utilizar contenedores, los cuales no se requieren pues no se propone migrar la aplicación hacia o desde otro sistema; por el contrario, se precisa que la aplicación corra de mejor forma en este equipo, por lo que será compilada aprovechando sus características y tratando de obtener un mejor rendimiento en el equipo.

Requisitos previos

Los requisitos básicos para la compilación de `OpenFOAM`, los cuales fueron instalados en el clúster `gissele`, son los siguientes:

- openmpi
- cmake
- qt
- procmail

```
yum install openmpi
yum install openmpi-devel
yum install cmake
yum install qt
yum install qt-devel
yum install procmail
```

Descarga del software

El software fue descargado de la siguiente página web:

`https://www.openfoam.com/download/install-source.php`

```
OpenFOAM-v1612+.tgz
ThirdParty-v1612+.tgz
```

Configuraciones

Fue añadida al `PATH` la siguiente ruta:

```
export PATH=$PATH:/usr/lib64/openmpi/bin:/usr/lib64/qt4/bin
```

Se modificó el archivo `/home/appl/OpenFOAM/OpenFOAM-v1612+/etc/bashrc`:

```
FOAM_INST_DIR=/home/appl/$WM_PROJECT
export WM_COMPILER=Gcc
export WM_ARCH_OPTION=64
export WM_MPLIB=SYSTEMOPENMPI
```

Se activaron las configuraciones en el ambiente:

```
source /home/appl/OpenFOAM/OpenFOAM-v1612+/etc/bashrc
```

Compilación

Compilación de las utilerías en el directorio ThirdParty-v1612+

ParaView

Utilería utilizada para post procesamiento de los resultados de OpenFOAM. La herramienta paraview fue un requerimiento original por parte de la unidad MOFABI, debido a ello se instaló, sin embargo, en la actualidad no se utiliza pues se prefiere transferir los archivos a

estaciones de trabajo con capacidades gráficas para visualizar los resultados. Aún así se ha documentado la instalación del software.

```
./makeParaView
```

Compilación OpenFOAM

Probar antes el sistema para verificar si está preparado:

```
foamSystemCheck
```

Cambiar al directorio principal $WM_PROJECT_DIR y ejecutar el siguiente comando:

```
Foam
```

Este comando probará todas las configuraciones realizadas.

Para aprovechar los recursos y realizar la compilación en paralelo, se definieron las siguientes variables de ambiente:

```
export WM_NCOMPPROCS=40
export WM_SCHEDULER=wmakeScheduler
export WM_HOSTS="gis2:20 gis3:20"
```

Después se realizó la siguiente compilación:

```
./Allwmake
```

Postcompilación

```
foamInstallationTest
```

Recursos adicionales

Swak4foam

Se instaló la extensión swak4foam utilizando el archivo openfoam-extend-swak4Foam-dev-branches-develop.tar.gz, el cual fue obtenido a partir de la versión de desarrollo.

```
git clone https://github.com/Unofficial-Extend-Project-
Mirror/openfoam-extend-swak4Foam-dev.git swak4Foam
git checkout branches/develop
```

La instalación se realizó en el directorio `/home/appl/swak4Foam`

```
./Allwmake > log.make 2>&1
```

DualSPHysics

El Código de Hidrodinámica de Partículas Suavizadas (*Smoothed Particle Hydrodynamis* o *SPH*, por sus siglas en inglés) desarrollado por las universidades de Vigo y de Manchester, el cual está basado en uno anterior llamado SPHysics, que fue elaborado a partir de un esfuerzo colaborativo entre dichas universidades y la Universidad Johns Hopkins (DualSPHysics, s.f.).

A diferencia de su antecesor, DualSPHysics permite realizar simulaciones de fluidos en superficies libres utilizando los procesadores CPU y GPU. Asimismo, el conjunto de códigos de DualSPHysics está desarrollado principalmente en C++ y CUDA.

Como ya se mencionó en los alcances de este proyecto, el clúster `gissele` no cuenta con tarjetas GPU, por lo cual se ha configurado para funcionar solamente en la modalidad de procesamiento en CPU.

Descarga del software

El software fue descargado de la siguiente página web:

`http://www.dual.sphysics.org/index.php/downloads`

```
DualSPHysics_v4.0_Linux_x64.tar.gz
```

Compilación

El software DualSPHysics se instaló en `/home/appl/DualSPHysics_v4.0_Linux_x64`

En el directorio
`/home/appl/DualSPHysics_v4.0_Linux_x64/SOURCE/DualSPHysics_v4/Source`

se compila mediante la opción “Sólo CPU”

```
make -f Makefile_cpu
```

Por omisión, en la compilación se suele utilizar la bandera `-f openmp`, debido a que este programa utiliza OpenMP para paralelizar el código.

5. Ejecución de pruebas de las aplicaciones

OpenFOAM

En el caso elaborado por un integrante de la Unidad MOFABI se realizó una simulación con las condiciones siguientes: fluido viscoso, agua (densidad constante) y estado transitorio. La velocidad de la corriente libre fue de 0.1 m/s. En cada iteración se resuelve la ecuación de continuidad y se obtiene el perfil de velocidades de las tres componentes (u, v, w). La velocidad de la corriente libre se encuentra en la dirección x, y se requiere analizar el efecto que tiene la capa límite en las direcciones (y, z).

Las primeras corridas ejecutadas con `OpenFOAM` se realizaron sin utilizar un sistema de administración de tareas. A continuación se detalla el proceso efectuado:

Antes de probar la aplicación, se añadió la configuración del ambiente:

```
# Configuracion para openfoam
module load mpi/openmpi-x86_64
source /home/appl/OpenFOAM/OpenFOAM-v1612+/etc/bashrc
alias wmUNSET='. $WM_PROJECT_DIR/etc/config/unset.sh'
```

El caso se corrió en el directorio `/datos/mofabi/COMPLETA-ORIFICIOS-2`.

Primero, se creó el archivo `nodos` donde se pondrán los nombres de los nodos y el número de cores en que se ejecutará el proceso, en este caso queda así:

```
gis4 cpu=20
gis5 cpu=20
```

Por lo tanto, la prueba se realizará utilizando 40 cores exclusivos.

La dimensión del problema (determinada previamente por el usuario, para compararla con los resultados obtenidos anteriormente) fue configurada en el archivo `system/decomposeParDict`, la cual debe coincidir con el número de *cores* que se van a utilizar. En este caso, el tamaño será dado por las líneas:

```
numberOfSubdomains 40;
simpleCoeffs
{
    n               ( 5 2 4 );
    delta           0.001;
}
```

La aplicación y el tiempo a simular se configuró en el archivo `system/controlDict`:

```
application     pisoFoam;
startFrom       startTime;
startTime       0;
stopAt          endTime;
endTime         20;
deltaT          0.0041666;
```

Lo anterior indica que se simularán 20 segundos del problema.

Preprocesamiento

El preprocesamiento se refiere al proceso que realiza la descomposición de dominio, dividiendo el trabajo que será enviado a los diferentes procesadores. Para eso se utiliza el siguiente comando:

```
decomposePar
```

El cual dará origen a los siguientes directorios:

```
processor0
processor1
…
processor39
```

Procesamiento

Para procesar el caso o llevar a cabo la simulación, OpenFOAM se ejecutó así:

```
nohup mpirun --hostfile /home/mofabi/COMPLETA-ORIFICIOS-2/nodos -
np 40 pisoFoam -parallel &
```

Postprocesamiento

Al finalizar, se le pide a OpenFOAM que reconstruya el caso mediante el comando:

```
reconstructPar
```

OpenFOAM con slurm

Para correr el caso, utilizando el sistema de administración de tareas, se realizó lo siguiente:

Primero se configuró el ambiente utilizando la herramienta *modules environment*, donde se definió el módulo de `OpenFOAM`:

Algunas de las líneas del módulo fueron las siguientes:

```
#%Module#####################################################################
###############
##
#
#  OpenFoam module for use with 'environment-modules' package:
#
proc ModulesHelp { } {
        global version
        puts stderr "\tCarga el ambiente para usar la aplicacion
OpenFOAM"
        puts stderr "\tOpenFOAM version v1612+\n"
        puts stderr "
     Version $version
"
}

prereq mpi/openmpi-x86_64
module-whatis   "\t Carga el ambiente para usar la aplicacion
OpenFOAMv1612+"

set        foamRoot   "/home/appl"

setenv          WM_PROJECT      "OpenFOAM"
setenv          WM_PROJECT_VERSION    "v1612+"
setenv          WM_COMPILER           "Gcc"
setenv          WM_COMPILE_OPTION     "Opt"
setenv     WM_ARCH               "linux64"
…
```

Se creó el archivo: `correcaso.job` para enviar el trabajo a su ejecución:

```
#!/bin/bash
#Nombre del Job
#SBATCH -J openfoam
#Archivo de salida estandar
#SBATCH -o correcaso.salida.%j
```

```
#Archivo de error estandar
#SBATCH -e correcaso.error.%j
#Cola
#SBATCH --partition=cola
#Numero de nodos
#SBATCH -N 2
#Numero de cores por nodo
#SBATCH --ntasks-per-node=40

#carga modulos
module load mpi/openmpi-x86_64
module load appl/OpenFoamv1612+

#Define ruta del caso y se cambia a ella
RUTA=/home/mofabi/COMPLETA-ORIFICIOS-2-g
cd $RUTA

#Corre el caso utilizando mpi en paralelo
#SLURM_NTASK=nodos*corespornodo
mpirun -np $SLURM_NTASKS pisoFoam -parallel
```

DualSPHysics

Antes de correr `DualSPHhysics` se configuró el directorio `/datos/RUN_DIRECTORY`, dentro del cual se pusieron los archivos de los datos de entrada de la prueba del software.

Para tal efecto se eligieron cinco casos:

```
1_CASEDAMBREAK/CaseDambreak_linux64_CPU.sh
2_CASEPERIODICITY/CasePeriodicity_linux64_CPU.sh
3_CASEMOVINGSQUARE/CaseMovingSquare_linux64_CPU.sh
4_CASEFORCES/CaseForces_linux64_CPU.sh
5_CASESLOSHING/ CaseSloshingAcc_linux64_CPU.sh
```

Posteriormente se creó un archivo que permitiera correr los casos, uno tras otro. En cada caso se utilizaron 20 cores, para lo cual se definió la variable `OMP_NUM_THREADS=20`.

6. Comparación de los resultados obtenidos

Las pruebas de ejecución de las aplicaciones realizadas en el clúster `gissele`, fueron realizadas también en la supercomputadora `Miztli`.

Condiciones de la prueba

	Gissele	Miztli
CPU	Intel Xeon E5-2680 v2 (2.8 GHz/10 core)	Intel Xeon E5-2660 v3 (2.6 GHz/10 core)
RAM	160 GB	128 GB
Sistema de archivos	gluster	lustre
Conectividad	10 GbE FLR-T	Infiniband QDR 40 Gbps
Sistema de Control de Tareas	No Ejecución directa en background con nohup	LSF
Sistema Operativo	Centos 7.3	RHEL 6.7
Compiladores	GNU 4.8.5	GNU 4.4.7 (OpenFoam) Intel 15.0.1 (DualSPHysics)
MPI	Openmpi 1.10.3	Intel MPI 5.0.2p

OpenFOAM

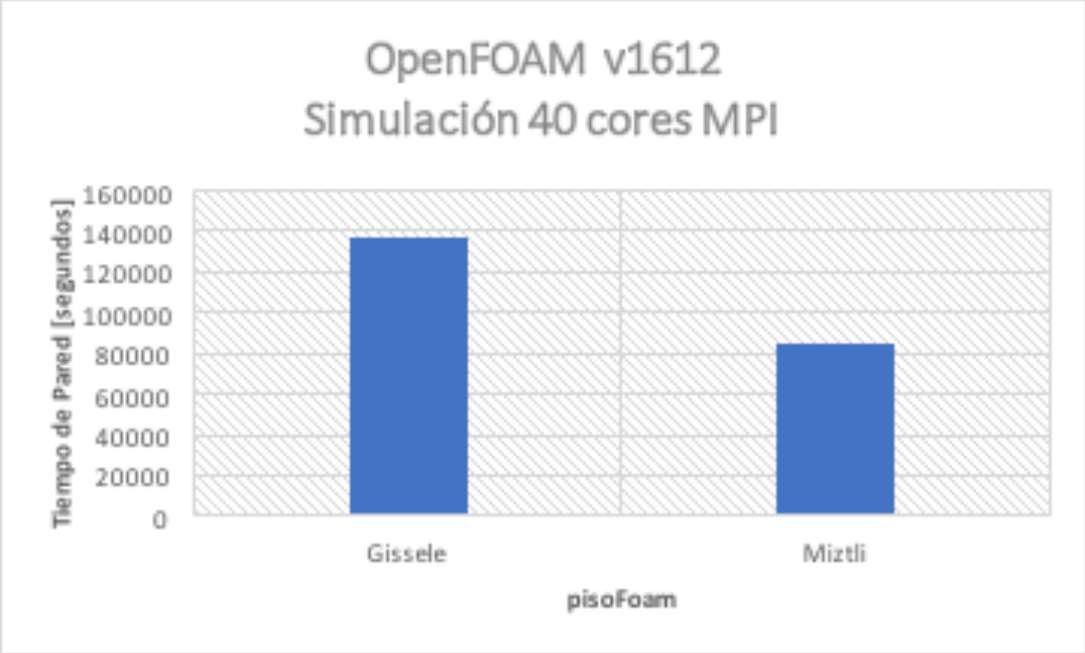


Figura 4. *Comparación del tiempo de pared en prueba OpenFOAM: clúster gissele vs. supercomputadora Miztli.*

Con los datos anteriores se obtuvieron los tiempos de ejecución de la etapa de procesamiento; la ejecución en dos nodos (40 *cores*) utilizando MPI, es la siguiente:

	Gissele [s]	Miztli [s]
pisoFoam	137428	84288

Al observar los tiempos de ejecución, nos damos cuenta de que Miztli corrió más rápido, pero eso puede ser debido a la velocidad en las comunicaciones y en los discos duros, así como en el rendimiento del sistema de almacenamiento.

DualSPHysics

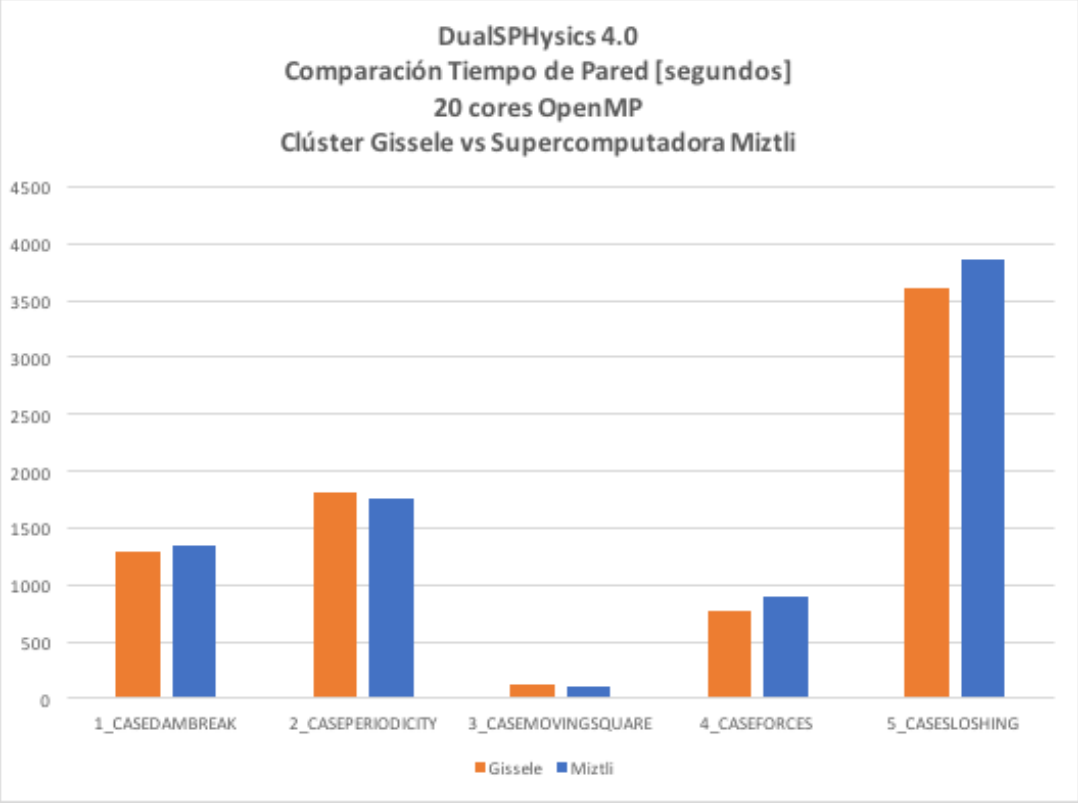


Figura 5. *Comparación del tiempo de pared en prueba DualSPHysics: clúster gissele vs. supercomputadora Miztli.*

Con los datos de la figura anterior, se obtuvieron los tiempos de ejecución de cada prueba. Un nodo (20 cores). OpenMP.

Casos	Gissele [s]	Miztli [s]
1_CASEDAMBREAK	1297.006958	1348.69751
2_CASEPERIODICITY	1805.869019	1761.249634
3_CASEMOVINGSQUARE	116.795723	98.556999
4_CASEFORCES	768.279236	895.37738
5_CASESLOSHING	3614.588623	3859.2771

En este caso se observa que los resultados son muy similares, favorecen ligeramente al clúster `gissele`; pero se debe tomar en cuenta que los procesadores son de mayor frecuencia.

7. Relación de las asesorías para el uso y mantenimiento del clúster

En esta etapa del proyecto se efectuaron diversas actividades, cuyo objetivo principal fue asesorar y capacitar a los integrantes de la unidad MOFABI en el uso y la administración del equipo, con el fin de mantener el clúster `gissele` operando de manera óptima.

Los temas abordados para cumplir con dicho objetivo, fueron los siguientes:

- Uso del sistema operativo GNU/Linux.
- Elementos básicos a considerar para implementar un clúster.
- Instalación del sistema operativo CentOS.
- Configuración básica del clúster.
- Curso básico de administración del sistema.
- Asesoría sobre los aspectos a considerar para elaborar e implementar las políticas para el uso eficiente del clúster.
- Instalación y uso de las aplicaciones `OpenFOAM` y `DualSPHysics`.

8. Propuestas futuras para el mejoramiento de la infraestructura

Como se mencionó anteriormente, en el análisis de la problemática existen algunas limitaciones para un funcionamiento óptimo del clúster `gissele`. Durante las etapas de desarrollo del proyecto se propusieron algunas adaptaciones para fortalecer su infraestructura como las siguientes:

1. Ampliar y mejorar la disponibilidad y redundancia en el sistema de almacenamiento mediante la adquisición de un sistema NAS, que pueda ser compartido por NFS con los nodos que conforman el clúster `gissele`.

2. Ampliar y mejorar la conectividad con la adquisición de un *switch* 10GbE, el cual será dedicado exclusivamente para el funcionamiento del clúster.

3. Realizar mejoras en la configuración del sistema de administración de tareas `Slurm`, que ya se encuentra instalado y configurado, pero no se ha extendido su uso a todos los usuarios. Para ello se requiere de un plan de capacitación, para que los miembros de la Unidad MOFABI hagan un mejor uso de Slurm.

Conclusiones

1. Existe una necesidad real, por parte de la Unidad de Modelación de Fluidos Ambientales, Biológicos e Industriales, en el uso de equipos HPC; sin embargo, al no haber tomado en cuenta todos los factores involucrados en la puesta a punto de un sistema de este tipo, dio como resultado que el clúster estuviera más de un año sin utilizar.

Con el desarrollo de este proyecto se logró integrar a la Unidad MOFABI una plataforma computacional que le permita realizar simulaciones en la dinámica de fluidos y en la transferencia de calor, utilizando las herramientas OpenFOAM y DualSPHysics.

El clúster gissele se encuentra operando desde el mes de abril de 2017, casi a su máxima capacidad.

2. Es conveniente difundir la importancia de una buena planeación y de contar con un plan integral de administración, el cual permita un uso adecuado de los sistemas HPC en la UNAM.

Se logró que tomaran conciencia tanto los miembros de la unidad MOFABI como el personal de cómputo del Centro de Ingeniería Avanzada, sobre la importancia de conocer las técnicas básicas de administración del sistema y de establecer las políticas que permitan el uso adecuado de los equipos.

3. La correcta administración de un equipo HPC requiere trabajar en conjunto con los usuarios, con el fin de ajustar el sistema a sus necesidades. En el caso de los miembros de la Unidad MOFABI, se contó con el apoyo necesario para lograr los objetivos deseados.

Durante el desarrollo de este proyecto, fueron apoyados en forma directa algunos estudiantes con diferente nivel educativo:

a) Licenciatura (3)
b) Maestría (5)
c) Doctorado (2)
d) Posdoctorado (1)

4. Es importante contar con técnicos especializados en la operación de los sistemas HPC, no solamente a nivel de usuario sino también en la administración. Recibieron capacitación dos miembros de la Unidad MOFABI, en la operación del sistema.

Referencias bibliográficas

Anoop, C. S. (2014) [Recuperado Agosto 2017], "Introduction to GlusterFS (File System) and Installation on RHEL/CentOS and Fedora", agosto de 2017, de TechMint. Sitio web: https://www.tecmint.com/introduction-to-glusterfs-file-system-and-installation-on-rhelcentos-and-fedora/

CFD Direct. (s.f.) (Recuperado en octubre de 2017). OpenFOAM. De CFD Direct. Sitio web: https://cfd.direct/openfoam/about/

Cisco (2008) [Recuperado abril 2017], "Protocolo Network Time Protocol: Informe oficial de mejores prácticas", de Cisco. Sitio web: https://www.cisco.com/c/es_mx/support/docs/availability/high-availability/19643-ntpm.html

Díaz, L. (2016). "2.2 Diseño y administración de clústers – Arquitectura y servicios", Apuntes de clase: Laboratorio de clústers y grids.

DualSPHysics, (s.f.) [Recuperado en octubre de 2017]. DualSPHysics. Sitio web: http://dual.sphysics.org/

Eijkhout, V. (2014), *Introduction to High Performance Scientific Computing*, E.U.A., Saylor.org Academy.

Flores, Y. y García, F. (2003), "Administración de tareas y balance de carga", Notas de taller, Semana de Supercómputo 2003, DGTIC, UNAM.

Hager, G. y Wellein, G. (2011), *Introduction to High Performance Computing for Scientists and Engineers*, E.U.A., CRC Press.

Levine, S. (2013) [Recuperado en junio de 2017]. Red Hat Global File System 2. De Red Hat Inc. Sitio web: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html-single/global_file_system_2/index

MOFABI (2017), *Quiénes somos*, mayo de 2017, de MOFABI. Sitio web: http://unidadmofabi.wixsite.com/mofabi-unam/quienes-somos1

Porkolab, Z. y Bear, M. (s.f.) [Recuperado Agosto 2017]. GlusterFS Storage Cluster on CentOS 7. De CentOS Community. Sitio web: https://wiki.centos.org/HowTos/GlusterFSonCentOS#head-dece0d4d435cf83fe10134d95aa0e5f39a13252d

Red Hat. (2009) [Recuperado en junio de 2017]. GFS 2 Sistema de Archivos Global. De Red Hat Inc. Sitio web: https://access.redhat.com/documentation/es-es/red_hat_enterprise_linux/5/html-single/global_file_system_2/index

Red Hat (2005) [Recuperado en mayo de 2017]. Sistema de archivos de red (NFS). De MIT. Sitio web: http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-nfs.html

SchedMD (2013) [Recuperado en mayo de 2018]. Slurm workload manager overview. De Sched MD. Sitio web: https://slurm.schedmd.com/overview.html

SchedMD (2018) [Recuperado en junio de 2018]. Slurm workload manager, Quick Start Administrator Guide. De Sched MD. Sitio web: https://slurm.schedmd.com/quickstart_admin.html