

Marcos Esteve Hernández

**AUTOMATIZACIÓN Y MONITORIZACIÓN DE UN SISTEMA PARALELO
DISTRIBUIDO BASADO EN ODROIDS**

TRABAJO DE FIN DE GRADO

dirigido por Carles Aliagas Castell

Grau d'Enginyeria Informàtica



UNIVERSITAT ROVIRA I VIRGILI

Tarragona

2021

Resum.

Aquest TFG recull el testimoni dels seus dos predecessors i busca unir-los, optimitzant i polint tant l'automatització com el monitoratge, i obtenir un sistema funcional que satisfaci els requisits d'un clúster de baix cost per a la computació en paral·lel. Es parteix d'un treball realitzat que es busca actualitzar, testejant totes les funcionalitats i detectant els errors que originen les noves versions de les eines, amb l'objectiu de corregir-los, aplicant una implementació el més general possible per minimitzar l'aparició de nous errors en un futur. A més a més, s'afegeixen funcionalitats, com un sistema de monitorització ja establert i amb més eines, així com scripts per a l'execució i el manteniment remot del clúster. El resultat és l'esperat, un sistema paral·lel distribuït funcional i automatitzat, amb marge de millora però sòlid en les seves funcions, que configura els nodes i la xarxa, permet l'execució en paral·lel i és fàcilment monitoritzable, mantenible i escalable.

Resumen.

Este TFG recoge el testigo de sus dos predecesores y busca aunarlos, optimizando y puliendo tanto la automatización como la monitorización, y obtener un sistema funcional que satisfaga los requisitos de un clúster de bajo coste para la computación en paralelo. Se parte de un trabajo realizado que se busca actualizar, testeando todas las funcionalidades y detectando los errores que originan las nuevas versiones de las herramientas, con el objetivo de corregirlos, aplicando una implementación lo más general posible para minimizar la aparición de nuevos errores en un futuro. Además, se añaden funcionalidades, como un sistema de monitorización ya establecido y con más herramientas, así como scripts para la ejecución y el mantenimiento remoto del clúster. El resultado es el esperado, un sistema paralelo distribuido funcional y automatizado, con margen de mejora, pero sólido en sus funciones, que configura los nodos y la red, permite la ejecución en paralelo y es fácilmente monitorizable, mantenible y escalable.

Abstract.

This Final Degree Project takes the baton from its two predecessors and seeks to bring them together, optimizing and polishing both the automation and monitoring, and to obtain a functional system that meets the requirements of a low-cost cluster for parallel computing. It starts from a work already carried out that seeks to update, to test all the functionalities and to detect the errors that originate the new versions of the tools, with the aim of correcting them, applying an implementation as general as possible to minimize the appearance of new errors in the future. In addition, functionalities are added, such as an already established monitoring system with more tools, as well as scripts for remote execution and maintenance of the cluster. The result is as expected, a functional and automated distributed parallel system, with room for improvement but solid in its functions, which configures the nodes and the network, allows parallel execution, and is easily monitorable, maintainable and scalable.

Índice

1	INTRODUCCIÓN	5
1.1	ANTECEDENTES.....	6
1.2	OBJETIVOS.....	7
2	HERRAMIENTAS DE TRABAJO.....	8
2.1	CONFIGURACIÓN DEL SISTEMA	8
2.1.1	<i>Ubuntu</i>	8
2.1.2	<i>Bash</i>	8
2.1.3	<i>SSH</i>	8
2.1.4	<i>DNSmasq</i>	9
2.1.5	<i>Iptables</i>	9
2.1.6	<i>NFS</i>	9
2.1.7	<i>Xfce</i>	10
2.1.8	<i>VNC</i>	10
2.1.9	<i>Slurm</i>	10
2.1.10	<i>Munge</i>	10
2.1.11	<i>Mpich</i>	11
2.1.12	<i>Midori</i>	11
2.2	MONITORIZACIÓN	11
2.2.1	<i>Ganglia</i>	11
2.2.2	<i>Apache</i>	11
2.2.3	<i>Monitorix</i>	12
3	REQUISITOS	13
3.1	REQUERIMIENTOS.....	13
3.2	CASOS DE USO	14
4	DISEÑO	19
4.1	DISEÑO DE LA ARQUITECTURA.....	19
4.1.1	<i>Diseño del software</i>	19
4.1.2	<i>Diseño de la red</i>	21
4.2	DECISIONES DE DISEÑO	22
4.2.1	<i>Para la configuración del máster</i>	22
4.2.2	<i>Para la configuración del slave</i>	23
4.2.3	<i>Herramienta de monitorización</i>	24
4.2.4	<i>Herramientas de administración remota</i>	25
5	IMPLEMENTACIÓN.....	26
5.1	CAMBIOS EN LOS SCRIPTS PARA LA CONFIGURACIÓN DEL MÁSTER.....	26
5.1.1	<i>init_master.sh</i>	26
5.1.2	<i>install_slurm.sh</i>	27
5.1.3	<i>add_monitoring.sh</i>	27
5.1.4	<i>Administración remota vía scripts</i>	28
5.2	CAMBIOS EN LOS SCRIPTS PARA LA CONFIGURACIÓN DE LOS NODOS SLAVES.....	29
5.2.1	<i>add_slave.sh</i>	29
5.2.2	<i>init_slave.sh</i>	30
5.3	SCRIPTS PARA LAS HERRAMIENTAS DE MONITORIZACIÓN	30
5.3.1	<i>Monitorix</i>	31
5.3.2	<i>Ganglia</i>	31
6	EVALUACIÓN.....	33
6.1	EVALUACIÓN DE LA CONFIGURACIÓN Y AUTOMATIZACIÓN DEL SISTEMA	33
6.1.1	<i>Evaluación del master</i>	33
6.1.2	<i>Evaluación del slave</i>	60
6.1.3	<i>Resultado final</i>	73

6.2	EVALUACIÓN DE LOS SISTEMAS DE MONITORIZACIÓN	75
6.2.1	<i>Ganglia</i>	75
6.2.2	<i>Monitorix</i>	79
6.3	EVALUACIÓN DEL SISTEMA DE ADMINISTRACIÓN REMOTA	81
6.3.1	<i>Vía interfaz de la página web interna</i>	81
6.3.2	<i>Vía scripts ubicados en /home/odroid/bin/</i>	82
7	PREVISIÓN DE FUTURO	84
8	CONCLUSIONES	85
9	ANEXOS	86
9.1	LISTADO DE VERSIONES	86
9.2	MANUAL DE INSTALACIÓN.....	87
9.2.1	<i>Paso 1: Cargar Imágenes</i>	87
9.2.2	<i>Paso 2: Montar Hardware</i>	87
9.2.3	<i>Paso 3: Instalar Software</i>	88
9.2.4	<i>Paso 4: Guía de Uso</i>	90
10	REFERENCIAS	92

Índice de tablas

TABLA 1. LISTADO DE VERSIONES DE PAQUETES UTILIZADOS EN LA CONFIGURACIÓN DEL SISTEMA.	86
--	----

Índice de figuras

FIGURA 1. DIAGRAMA DE CASOS DE USO DEL SISTEMA.	14
FIGURA 2. DIAGRAMA DE SCRIPTS DE LA INICIALIZACIÓN DEL MÁSTER.....	20
FIGURA 3. DIAGRAMA DE SCRIPTS DE LA INICIALIZACIÓN DE UN NODO SLAVE.....	21
FIGURA 4. DISEÑO DE LA RED EN LA QUE SE HA ELABORADO ESTE TFG.....	22
FIGURA 5. PRUEBA DE LA CORRECTA CONFIGURACIÓN DEL SLURM.....	73
FIGURA 6. COMPILACIÓN DE SCRIPTS PARA LA EVALUACIÓN DEL SISTEMA.....	74
FIGURA 7. EJECUCIÓN DEL SCRIPT MD5K_MPI.C CON 2 Y 4 THREADS.....	74
FIGURA 8. EJECUCIÓN DEL SCRIPT KMEAN_MPI.C CON 2, 4 Y 8 THREADS.....	75
FIGURA 9. PÁGINA INICIAL DE LA HERRAMIENTA GANGLIA.....	76
FIGURA 10. GRÁFICOS DE LAS MÉTRICAS DEL NODO MÁSTER EN LA PÁGINA DE GANGLIA (1).....	76
FIGURA 11. GRÁFICOS DE LAS MÉTRICAS DEL NODO MÁSTER EN LA PÁGINA DE GANGLIA (2).....	77
FIGURA 12. GRÁFICOS DE LAS MÉTRICAS DEL NODO SLAVE 'ODROID1' EN LA PÁGINA DE GANGLIA EN EL MOMENTO DE INICIALIZAR EL NODO.....	77
FIGURA 13. GRÁFICOS DE LAS MÉTRICAS DEL NODO SLAVE 'ODROID1' TRAS LA EJECUCIÓN DEL CÓDIGO MD5K_MPI.C....	78
FIGURA 14. GRÁFICOS DE LAS MÉTRICAS DEL NODO SLAVE 'ODROID1' TRAS LA EJECUCIÓN DEL CÓDIGO KMEAN_MPI.C...	78
FIGURA 15. VISIÓN DE LA VENTANA DE AUTENTICACIÓN AL CONECTAR CON LA PÁGINA DE MONITORIX.....	79
FIGURA 16. PÁGINA PRINCIPAL DE LA HERRAMIENTA MONITORIX.....	80
FIGURA 17. GRÁFICOS DEL NODO MÁSTER DESDE LA PÁGINA DE MONITORIX (1).....	80
FIGURA 18. GRÁFICOS DEL NODO MÁSTER DESDE LA PÁGINA DE MONITORIX (2).....	81
FIGURA 19. PRUEBA DEL SISTEMA DE ADMINISTRACIÓN REMOTA MEDIANTE LA INTERFAZ WEB (1).....	82
FIGURA 20. PRUEBA DEL SISTEMA DE ADMINISTRACIÓN REMOTA MEDIANTE LA INTERFAZ WEB (2).....	82
FIGURA 21. PRUEBA DE LA EJECUCIÓN DEL SCRIPT DE ADMINISTRACIÓN REMOTA EN PARALELO (1).....	83
FIGURA 22. PRUEBA DE LA EJECUCIÓN DEL SCRIPT DE ADMINISTRACIÓN REMOTA EN SECUENCIAL (1).....	83
FIGURA 23. PRUEBA DE LA EJECUCIÓN DEL SCRIPT DE ADMINISTRACIÓN REMOTA EN SECUENCIAL (2).....	83
FIGURA 24. PRUEBA DE LA EJECUCIÓN DEL SCRIPT DE ADMINISTRACIÓN REMOTA EN PARALELO (2).....	83

1 Introducción

Hoy en día, gracias al avance de las tecnologías, la cantidad de información que se tiene disponible es inmensa. En algunos ámbitos, podemos hablar de algoritmos que trabajan con gigas y gigas de información, que iteran millones de veces para procesar estos datos, y esto, claro está, conlleva un coste. Y cuando hablamos de coste, lo hacemos en términos de dinero, por supuesto, pero también de tiempo. Procesar toda esta información requiere de tecnologías que se encuentren a la vanguardia en cuanto a potencia se refiere, pero aún así, solo con eso no es suficiente. Siempre, en todo algoritmo, en toda ejecución, llega un punto en el que no se puede hacer más rápido u óptimo, hay un coste temporal que no se puede reducir, aunque tengas el procesador más potente. Y es por ello por lo que surge la computación en paralelo. Ésta no deja de ser una estrategia que sigue uno de los principios más antiguos que existen: ‘divide y vencerás’. La computación en paralelo nace con el objetivo de reducir el coste temporal de los algoritmos, de poder ejecutar fragmentos de código al mismo tiempo, optimizando al máximo los recursos para que, en todo momento, dentro de lo posible, se estén aprovechando al máximo las capacidades de nuestro sistema. Es de esta manera que surgen nuevos modelos de computadores, los multicomputadores. La premisa de estos multicomputadores es la de tener una gran cantidad de computadores individuales, cada uno de ellos con sus procesadores, su espacio de memoria, ..., pero que se encuentren conectados entre sí para un fin común. De esta forma, se busca que todos estos computadores se comporten como si fueran uno solo. A este conjunto se le denomina clúster. Y tener un clúster aporta muchas ventajas, por ejemplo, en la escalabilidad. Al estar compuesto por muchos computadores individuales, si queremos aumentar el tamaño de nuestro clúster, no tendremos más que añadir más computadores al sistema, conectarlos a la red y configurarlos de manera que sean parte del clúster. De esta forma, podemos contar con una gran cantidad de procesamiento, centrada en una tarea común, como si de un único ordenador se tratase, y así hacer frente a los nuevos retos tecnológicos que nos sobrevienen.

El proyecto en el que este Trabajo Final de Grado se centra en hacer frente a esos costes que hemos comentado, temporal y monetario, utilizando la premisa de la computación en paralelo y los clústeres. El objetivo de éste es proporcionar un sistema que sea capaz de ejecutar código en paralelo con altos niveles de escalabilidad, y que a su vez esto no implique un gasto excesivo y no se quede en un mero proyecto teórico, queremos que sea real. Por ello, la propuesta que realizaron Carles Aliagas et al. en su artículo [1] es la de implementar un ‘minimulticomputador’ basado en placas Odroid C2. Estas placas nos permiten tener un computador con unas características mínimas a un coste reducido. De esta manera, se podría implementar un clúster con unas características importantes, fruto de la combinación de estas placas, a un precio no tan desorbitado como pueden alcanzar algunos multicomputadores del mercado.

La idea de este trabajo es la de proveer, a cualquiera que necesite o quiera montar un sistema distribuido para la computación en paralelo basado en placas Odroid, un software que le dé esta posibilidad. Y eso es lo que se ha realizado en este trabajo, un conjunto de scripts que instalan y configuran todos los aspectos del sistema para obtener un sistema distribuido, en el que los nodos están interconectados mediante Ethernet formando una LAN¹, y en la que se establecen dos jerarquías: una de las placas ejerce como ‘master’,

¹ LAN: *Local Area Network*, es una red de computadores que abarca un área reducida.

controla y monitoriza a las demás, y es la que manda al resto, a los nodos ‘slave’, lo que deben ejecutar. De esta forma, se facilita la comunicación en la red y se trabaja como si de un único computador se tratase.

1.1 Antecedentes

Este trabajo recoge el hilo de dos Trabajos Finales de Grado previos y los aúna, con el objetivo de obtener una versión actualizada, pulida y funcional del sistema de automatización y monitorización del clúster.

El primer TFG es el trabajo desarrollado por David Ferrer Fernández “Automatització de la creació d’un servidor paral·lel distribuït basat en Odroids”. En su trabajo, David realiza una enorme labor estableciendo las tecnologías que se utilizan, tanto a nivel software como hardware, y las cuales en su gran mayoría son las mismas que utilizo yo en mi trabajo, además de implementar toda una jerarquía de scripts en lenguaje bash para automatizar el proceso de creación del sistema y la adición de nodos al clúster. Para conseguir este objetivo, emplea diversos paquetes para proveer los servicios esenciales para una LAN, como es el servicio de DNS, de DHCP, un firewall, SSH para la conexión entre nodos, ..., así como de otros que no son estrictamente necesarios para la configuración de la red, pero que sí suponen una ventaja cuando hablamos de sistemas distribuidos, como es el NFS para la compartición de archivos y directorios, Munge como servicio de autenticación, o SLURM, un software que proporciona herramientas muy útiles en un clúster como el acceso excluyente a los nodos, la ejecución de tareas o un sistema de colas de trabajos pendientes. Todos estos servicios serán tratados un poco más en profundidad durante la memoria de este trabajo, explicando los cambios y las adaptaciones que han sido necesarios en ellos para que sean funcionales en la configuración del sistema final. Pero era necesario remarcar que gran parte de la autoría de estos scripts son de David y que mi trabajo parte de lo que él realizó.

El segundo de los trabajos predecesores al mío fue desarrollado por Joan Jara Bosch, y tiene como título “Monitorització d’un servidor paral·lel distribuït basat en plaques Odroid”. La parte del proyecto que correspondió a Joan es la de establecer un sistema de monitorización para el clúster. Para ello, diseñó e implementó una aplicación web que permitía la monitorización y el mantenimiento del sistema. En la página web, ubicada de forma local en la placa ‘master’ del sistema, se puede encontrar información sobre el proyecto y sus integrantes, un apartado de monitorización de las métricas del clúster y un sistema basado en la ejecución de comandos vía SSH para llevar a cabo un mantenimiento, al menos mínimo, del sistema, de una manera rápidamente accesible y con la ventaja que aporta al usuario el hecho de que sea mediante una interfaz gráfica. Durante el desarrollo de mi trabajo y en la versión final, se ha mantenido casi la totalidad de lo realizado por Joan, pero debido a las decisiones que se han tomado, la única parte que, más o menos, tendrá algo de relevancia de éste en la versión definitiva es el apartado de mantenimiento, ya que en mi trabajo se configura una herramienta para la monitorización del sistema que cuenta con más funcionalidades, por lo que el anterior sistema de monitorización de Joan no ha sido utilizado.

1.2 Objetivos

Este trabajo lo he desarrollado teniendo presentes diversos objetivos. A nivel académico y como mi Trabajo Final de Grado que es, mi ambición con este proyecto es la de profundizar de una manera más aplicada, en un proyecto real, y aplicar los conocimientos adquiridos durante la carrera, sobre todo en ámbitos como la administración de sistemas y redes, la computación paralela, la arquitectura de computadores y los sistemas distribuidos. A nivel de desarrollo del sistema como proyecto en sí y no únicamente como trabajo universitario, quería ser capaz de obtener un software que fuera realmente funcional y con el que cualquier persona que lo utilice pueda configurar su propio sistema paralelo distribuido, para poder hacer un uso real del mismo, no como muchos trabajos que he realizado durante la carrera, en los que no había un objetivo real detrás del programa que yo hacía, nadie lo iba a utilizar y sólo servía para aprobar la asignatura o sacar mejor nota.

2 Herramientas de trabajo

En esta sección proporciono una breve descripción de los componentes software que he utilizado en el proyecto y que proporcionan las funcionalidades que el sistema paralelo distribuido requiere y ofrece.

2.1 Configuración del sistema

2.1.1 Ubuntu

Ubuntu es el sistema operativo de las placas Odroid del clúster. Es un sistema de software libre y código abierto, una distribución de GNU/Linux² basada en Debian³, muy utilizado tanto en computadores de escritorio como en servidores.

Se ha escogido Ubuntu como sistema operativo, ya que se adapta perfectamente a las características de las placas Odroid C2, tanto que desde la página web oficial de éstas se recomienda utilizar Ubuntu. Para este proyecto se utiliza la versión Ubuntu 20.04 Minimal, por razones que se detallan en la sección de Diseño de esta memoria.

2.1.2 Bash

Bash es una interfaz de usuario de línea de comandos, una Shell⁴ de Unix, además de un lenguaje de scripting. Generalmente, se encuentra como una ventana de texto donde el usuario introduce comandos por teclado para comunicarse con el núcleo del sistema y ejecutar herramientas que le permiten controlar el computador. Además, es capaz de leer y ejecutar comandos desde un archivo, es decir, se pueden implementar scripts para que el intérprete los ejecute. Bash es el intérprete de comandos que viene por defecto en la mayoría de distribuciones GNU/Linux, por lo que es el que se ha utilizado para desarrollar este proyecto.

2.1.3 SSH

SSH (Secure SHell) es un protocolo y un programa que permite el acceso remoto a otro dispositivo de una manera segura, creando un túnel en el que la información intercambiada entre los dos extremos se encuentra cifrada. Este túnel permite tanto el acceso entre computadores como también el paso de información, ya sea el paso de ficheros, directorios o ejecutar tanto comandos como scripts de manera remota. Un aspecto clave para el proyecto que nos ofrece SSH es la capacidad de compartir claves RSA entre los distintos nodos de la red, lo que nos facilita la ejecución y la administración remota en toda la red, sin la necesidad de usar las contraseñas de los usuarios cada vez que queremos conectarnos a un nodo.

En el proyecto se utiliza tanto para el acceso e instalación de las placas, como para la ejecución de comandos y scripts para tareas de administración y mantenimiento.

² GNU/Linux: es una denominación que reciben los sistemas operativos tipo Unix y que se caracteriza por tener el núcleo del sistema operativo Linux junto con el entorno GNU, un conjunto de paquetes software.

³ Debian: es un sistema operativo libre GNU/Linux.

⁴ Shell: es un término utilizado para referirse a un intérprete de comandos.

2.1.4 DNSmasq

DNSmasq es una herramienta que cumple una doble función en el sistema: por un lado, proporciona un servicio de DNS⁵, respondiendo a las peticiones, o bien mediante su pequeña caché local (que utilizaremos principalmente para los dispositivos de la red interna) o bien redireccionando estas peticiones a un servidor real y recursivo. Por otro lado, esta herramienta incorpora un pequeño servidor de DHCP⁶ que soporta tanto peticiones estáticas como dinámicas y que utilizaremos para configurar la red interna, donde se encuentran los nodos de nuestro clúster.

Se utiliza la herramienta `dnsmasq` en el sistema, ya que tiene bajos requerimientos, lo cual es algo que buscamos en todos los componentes de nuestro sistema, debido a que las placas no tienen unas capacidades muy elevadas, además de porque proporciona en una sola herramienta dos servicios que son necesarios para implementar un clúster.

2.1.5 Iptables

Iptables es una herramienta que permite modificar fácilmente las tablas del firewall o cortafuegos de Linux. Estas tablas contienen un conjunto de políticas y reglas que regulan el paso de paquetes de red, los que entran, los que salen, y los que pasan a través. La configuración típica de estas tablas es la de aplicar políticas restrictivas y a continuación permitir el paso solo de los paquetes que estén autorizados; de esta manera se tiene un mayor control sobre la red.

En este sistema se implementan las reglas iptables en el nodo master, que actúa como firewall entre la red interna e Internet. Se aplican políticas restrictivas y se permite únicamente el paso de paquetes mediante las vías que utilizan las herramientas configuradas y el sistema.

2.1.6 NFS

NFS (*Network File System*) es un protocolo a nivel de aplicación que implementa un sistema de archivos distribuido en una LAN, es decir, permite la compartición de archivos y directorios de manera simultánea y síncrona entre los nodos del clúster.

Esta herramienta es muy útil en nuestro sistema, en el que todos los nodos comparten el mismo directorio 'home' ubicado en el master. Esto facilita la ejecución de scripts en toda la red, el paso y el acceso a los archivos, y garantiza que todos los nodos dispongan de las versiones actualizadas y consistentes de los ficheros de configuración, como en el caso del fichero de configuración de Slurm.

⁵ DNS: *Domain Name System*, es un sistema de nomenclatura jerárquico en el que principalmente se asocia una información y un nombre a una dirección IP.

⁶ DHCP: *Dynamic Host Configuration Protocol*, es un protocolo de red del tipo cliente/servidor (donde el servidor será el master y los clientes los nodos slave) cuya principal función es asignar una dirección IP y otros parámetros de configuración a un dispositivo para que pueda comunicarse por la red.

2.1.7 Xfce

Xfce es un entorno de escritorio libre utilizado en sistemas GNU/Linux como el nuestro. Proporciona una implementación gráfica de un entorno de escritorio, de manera que, si conectamos nuestra placa máster a una pantalla, podamos visualizar un entorno gráfico como con los que estamos acostumbrados a trabajar hoy en día y no únicamente un terminal de comandos de texto.

2.1.8 VNC

VNC (*Virtual Network Computing*) es un programa de software libre que implementa un pequeño servidor que permite observar, desde un ordenador cliente remoto, la pantalla del ordenador que hace la función de servidor. Esta característica, en conjunto con Xfce, nos permite la posibilidad de visualizar y controlar el nodo máster de nuestro sistema desde un ordenador externo, de manera remota.

2.1.9 Slurm

Slurm (*Simple Linux Utility for Resource Management*) es un software de gestión de tareas y clústeres. Es una herramienta que, pese a no ser imprescindible en un clúster, proporciona una serie de funcionalidades que son necesarias si se busca una optimización del sistema, como son: implementa un sistema de colas tanto para usuarios como para tareas o procesos, de manera que permite el acceso de manera controlada y excluyente a los recursos del sistema; y además proporciona herramientas para iniciar, ejecutar y supervisar los trabajos y acciones sobre los nodos.

Slurm implementa un sistema con dos tipos de actores: el controlador o administrador, y el esclavo. El controlador, ubicado en el máster, tiene conocimiento del estado y características del sistema, y es el encargado de gestionar cómo se reparten los recursos y garantizar que se están aprovechando al máximo de sus capacidades. Por otro lado, los esclavos tienen como única tarea prestar sus capacidades para la ejecución de lo que les asigna el controlador y devolver el resultado, despreocupándose del estado del sistema y de cualquier tarea que no sea la suya. Junto a todo esto, Slurm es capaz de seguir diferentes políticas en cuanto a la asignación de recursos, siendo la predeterminada la FIFO⁷, e implementa un sistema de estados para el seguimiento de las tareas o trabajos.

2.1.10 Munge

Munge es un servicio de autenticación para crear y validar credenciales, con el cual, mediante la creación y compartición de una clave simétrica, permite a Slurm autenticar y validar la identidad de los nodos de la red. Además, cuenta con un diseño ampliamente escalable pensado para trabajar con entornos como nuestro sistema paralelo distribuido.

⁷ FIFO: *First In First Out*, política que se basa en la premisa que el primero en requerir de un recurso es al que se le otorga y lo mantiene hasta terminar su tarea.

2.1.11 *Mpich*

Mpich es un software libre que implementa el estándar MPI⁸. Se utiliza para el paso de mensajes en programas que se ejecutan en entornos de memoria distribuida en computación paralela. Se trata de una alternativa a OpenMPI, utilizada en la asignatura de Computación Paralela y Masiva, que se ha añadido al proyecto únicamente debido a problemas de instalación y configuración de este último.

2.1.12 *Midori*

Midori es un navegador web ligero, una característica importante en nuestro sistema, que se instala en la placa máster y se utiliza principalmente para el acceso a las páginas de monitorización (en formato html). A éstas se debe acceder mediante un navegador ubicado en la placa máster, ya que desde un dispositivo que no se encuentre en la red interna no es posible visualizar las métricas que los nodos envían y que recogen estas herramientas de monitorización.

2.2 Monitorización

2.2.1 *Ganglia*

Ganglia es una herramienta de monitorización distribuida y escalable pensada para entornos de sistemas distribuidos y clústeres. Es un sistema que permite la visualización de las métricas tanto en vivo como en grabación. Es un sistema que utiliza tecnologías como RRDtool⁹ para almacenar y visualizar las métricas o XML¹⁰ para la representación de los datos. Está organizado en dos servicios: una aplicación web basada en PHP¹¹, que hace a su vez de servidor, y un servicio ‘cliente’ ubicado en cada nodo, encargado de recoger y enviar las métricas del nodo al servidor.

2.2.2 *Apache*

Apache es un servidor web HTTP¹² de código abierto utilizado en sistemas GNU/Linux. Ganglia necesita que esté configurado este servidor para poder alojar su página web, que permite al usuario visualizar las métricas del sistema y de la red.

⁸ MPI: *Message Passing Interface*, es un estándar que implementa una biblioteca de paso de mensajes diseñada para programas que se ejecutan en multicomputación.

⁹ RRDtool: *Round Robin Database tool*, es una herramienta para el tratamiento de datos que trabaja con bases de datos y que lleva a cabo una planificación *Round Robin* a la hora de manejar los datos. Se utiliza principalmente para datos temporales y seriales, como las métricas del clúster.

¹⁰ XML: *eXtensible Markup Language*, es un metalenguaje que permite almacenar datos de forma legible en formato texto.

¹¹ PHP: es un lenguaje de programación utilizado principalmente para desarrollo web.

¹² HTTP: *HyperText Transfer Protocol*, es un protocolo que permite la transferencia de información a través de Internet, ampliamente utilizado para la comunicación entre los navegadores y los servidores web.

2.2.3 *Monitorix*

Monitorix es una herramienta ligera para la monitorización de sistemas. Es un software libre diseñado para sistemas GNU/Linux y clústeres. Monitorix incluye su propio servidor HTTP, ubicado por defecto en el puerto TCP¹³ 8080, donde se pueden visualizar las métricas y la carga de red del sistema.

¹³ TCP: *Transmission Control Protocol*, es uno de los protocolos principales de Internet, el cual implementa el intercambio seguro y controlado de paquetes de datos.

3 Requisitos

3.1 Requerimientos

Para poder sacar este proyecto adelante he tenido que verificar que se cumplieran una serie de requerimientos que debía tener el sistema y éstos debían ejecutarse correctamente, para garantizar que las funcionalidades que ofrece el clúster se veían satisfechas. A continuación, los enumero:

- Correcta configuración de la LAN.
 - Servicio DHCP en la red interna.
 - Servicio DNS en la red interna.
 - Acceso a Internet desde la red interna.
 - Seguridad mediante *firewall*.
 - Acceso a los nodos de la red sin necesidad de contraseña (mediante una clave pública).
 - Compartición y acceso a ficheros y directorios comunes.
- Correcta configuración de Slurm para implementar un sistema de colas para diversos usuarios y trabajos, capacidad de ejecutar tareas utilizando los nodos del clúster, paso de mensajes.
- Permitir la modificación de características del sistema como el usuario por defecto, la zona horaria, el idioma del sistema, el rango de direcciones de la red interna, etc.
- Proveer de un sistema de monitorización del sistema, que permita visualizar las métricas y uso de la red tanto de la placa máster como del resto de placas slave.
- Proveer de herramientas para la administración remota del sistema, para llevar a cabo tareas de mantenimiento y solución de posibles fallos.
- Ofrecer una vía automatizada de instalación y configuración del sistema.
- Escalabilidad, facilidad para incorporar y configurar nuevos nodos al clúster.
- Proporcionar manuales para guiar a los usuarios, tanto en el proceso de configuración del sistema como en su uso habitual.

3.2 Casos de uso

Los casos de uso son descripciones de una acción o proceso que se puede llevar a cabo, en este caso, en nuestro sistema. Éstos ofrecen información sobre cómo se comporta el sistema, o sobre las acciones que realizan los usuarios, los actores. En nuestro sistema, pueden intervenir dos tipos de actores: el administrador y el usuario. A continuación, se muestra el diagrama de casos de uso para nuestro sistema.

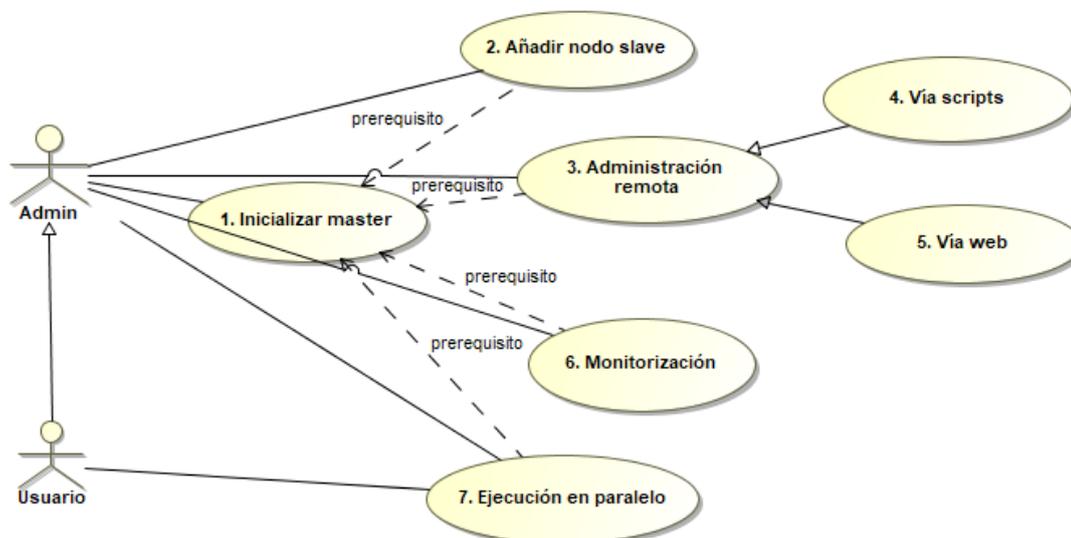


Figura 1. Diagrama de casos de uso del sistema.

Caso de uso 01. Inicializar máster

Resumen de la acción: Instala y configura el nodo máster.

Parámetros de entrada: Ninguno.

Parámetros de salida: Ninguno.

Actores: Administrador.

Precondición: Placas y hardware conectado.

Postcondición: Nodo máster y nodos slaves conectados correctamente configurados.

Proceso principal:

1. El Administrador tiene el fichero comprimido con el software.
2. El Administrador establece la conexión con el comando `ssh-keygen -R direcciónIP`.
3. El Administrador envía mediante `scp` el fichero a `root@direccionIP:/root`
4. El Administrador se conecta a la placa vía `ssh root@direccionIP`.
5. El Administrador descomprime el fichero comprimido mediante el comando `tar xzvf odroid-cluster.tgz`
6. El Administrador se desplaza al directorio `odroid-cluster` mediante el comando `cd odroid-cluster`

7. El Administrador ejecuta el script de configuración del máster mediante el comando `./init_master.sh`
8. El sistema inicia la configuración automática.
9. El Administrador interactúa brevemente con el proceso de instalación (establecer contraseñas, darle a aceptar, etc.).
10. El sistema se configura correctamente.

Alternativas del proceso y excepciones:

3a. El Administrador se conecta directamente a la placa, conectándola a una pantalla.

Caso de uso 02. Añadir nodo slave

Resumen de la acción: Se añade un nuevo nodo al clúster y se configura.

Parámetros de entrada: Nombre del nuevo nodo.

Parámetros de salida: Ninguno.

Actores: Administrador.

Precondición: El máster ha sido inicializado y hay al menos una placa más conectada.

Postcondición: Se ha añadido y configurado un nuevo nodo *slave*.

Proceso principal:

1. El Administrador se encuentra como usuario root en el directorio `/root/odroid-cluster`.
2. El Administrador ejecuta el comando `./add_slave.sh odroidX`, siendo X el número correspondiente al nodo que se quiere añadir al clúster.
3. El sistema comienza la configuración del nuevo nodo.
4. El sistema requiere de las contraseñas del usuario por defecto del sistema y del usuario root de los nodos *slave*.
5. El Administrador introduce las contraseñas que el sistema ha pedido.
6. El sistema configura correctamente el nodo odroidX.

Alternativas del proceso y excepciones:

Caso de uso 03. Administración remota

Resumen de la acción: Ejecutar comandos en los diferentes nodos del clúster para llevar a cabo tareas de mantenimiento y solución de errores.

Parámetros de entrada: Ninguno.

Parámetros de salida: Ninguno.

Actores: Administrador.

Precondición: El nodo máster está inicializado.

Postcondición: Se ejecutan los comandos en los nodos seleccionados.

Proceso principal:

1. El Administrador decide si quiere usar los scripts ubicados en /home/odroid/bin o usar la herramienta de mantenimiento de la web local del máster.

Alternativas del proceso y excepciones:

Caso de uso 04. Administración remota vía scripts

Resumen de la acción: Ejecutar comandos en los diferentes nodos del clúster para llevar a cabo tareas de mantenimiento y solución de errores desde el terminal por medio de scripts.

Parámetros de entrada: Ninguno.

Parámetros de salida: Ninguno.

Actores: Administrador.

Precondición: El nodo máster está inicializado.

Postcondición: Se ejecutan comandos en los nodos del clúster.

Proceso principal:

1. El Administrador abre un terminal.
2. El Administrador ejecuta el comando `global_execute_par` seguido del comando que quiere ejecutar en paralelo en los nodos del clúster.
3. El sistema comprueba los nodos disponibles en el fichero `/opt/scripts/odroid.par`.
4. El sistema lanza la ejecución del comando introducido por el Administrador en todos los nodos disponibles.
5. El sistema muestra el resultado de la ejecución en cada nodo conforme van terminando.

Alternativas del proceso y excepciones:

- 2a. El Administrador ejecuta el comando `global_execute_seq` seguido del comando que quiere ejecutar en secuencial en los nodos del clúster.
- 5a. El sistema muestra el resultado de la ejecución en cada nodo de manera secuencial.

Caso de uso 05. Administración remota vía web

Resumen de la acción: Ejecutar comandos en los diferentes nodos del clúster para llevar a cabo tareas de mantenimiento y solución de errores, desde la herramienta que proporciona la web interna de la placa máster.

Parámetros de entrada: Ninguno.

Parámetros de salida: Ninguno.

Actores: Administrador.

Precondición: El nodo máster está inicializado.

Postcondición: Se ejecutan comandos en los nodos del clúster seleccionados.

Proceso principal:

1. El Administrador se abre un navegador web y se dirige a la página <https://direccionIPmaster>
2. El Administrador clic en la sección *Manteniment*.
3. El sistema muestra una pantalla para iniciar sesión.
4. El Administrador introduce usuario y contraseña.
5. El sistema muestra la página de mantenimiento.
6. El Administrador selecciona los nodos a los que se quiere conectar y clic en el botón Conectar.
7. El Administrador escribe un comando en la barra de texto y clic en el botón 'Enviar comanda'.
8. El sistema muestra el resultado del comando por el terminal de salida.

Alternativas del proceso y excepciones:

- 8a. El sistema muestra el resultado del comando por el terminal de errores.

Caso de uso 06. Monitorización

Resumen de la acción: Visualizar las métricas y el uso de la red del sistema.

Parámetros de entrada: Ninguno.

Parámetros de salida: Ninguno.

Actores: Administrador.

Precondición: El nodo máster está inicializado.

Postcondición: Se muestran las métricas en forma de gráficos, según las opciones seleccionadas.

Proceso principal:

1. El Administrador abre un navegador desde la placa máster.
2. El Administrador se dirige a la dirección <http://localhost/ganglia>
3. El sistema muestra la página principal de Ganglia.
4. El Administrador selecciona el nodo que desea visualizar.
5. El sistema muestra los gráficos de las métricas del nodo seleccionado.

Alternativas del proceso y excepciones:

Caso de uso 07. Ejecución en paralelo

Resumen de la acción: Se ejecuta un script utilizando los nodos del clúster.

Parámetros de entrada: Script a ejecutar.

Parámetros de salida: Resultado del script.

Actores: Administrador o Usuario.

Precondición: El nodo máster está inicializado.

Postcondición: Se ha ejecutado un script utilizando los nodos del clúster.

Proceso principal:

1. El actor ejecuta un script mediante el comando `srun`, añadiendo las opciones que desee.
2. El sistema informa que el proceso se ha creado correctamente.
3. El actor puede consultar el estado del proceso que ha mandado ejecutar mediante el comando `squeue`.
4. El sistema muestra el resultado del proceso ejecutado utilizando los nodos del clúster.

Alternativas del proceso y excepciones:

- 2a. El sistema muestra un mensaje de error.

4 Diseño

4.1 Diseño de la arquitectura

La arquitectura de este sistema se puede describir como el conjunto de dos arquitecturas: el diseño del software e interacción entre scripts para configurar el sistema, y la arquitectura de la red y las conexiones entre los distintos componentes del sistema.

4.1.1 Diseño del software

Dentro de la arquitectura del software, también podríamos definir dos subgrupos en relación con las tareas que se llevan a cabo y el resultado que generan.

El primero de ellos es todo lo relativo a la configuración del nodo máster y las herramientas que manejarán la red. El primer paso en la configuración del sistema es inicializar el nodo máster, instalando y configurando los paquetes necesarios. Para esto, la decisión desde un inicio fue la de centralizarlo todo bajo la ejecución de un único script. De esta manera, cuando el administrador ejecuta el script *init_master.sh*, pone en funcionamiento el código de éste para instalar los paquetes necesarios, crear el usuario odroid (por defecto) y cambiar la contraseña del usuario root, configurar aspectos como el NFS o Ganglia, instalar la interfaz gráfica, etc. Pero también este script realiza llamadas a otros scripts: *conf_network_master.sh* junto con *iptables.sh* se encargan de establecer las interfaces del sistema, de poner al máster en la lista de hosts conocidos, de activar el *forwarding* y de establecer las reglas del firewall para permitir y denegar el paso de paquetes; *dhcp_script.sh* contiene un conjunto de funciones que utiliza el servidor *dhcp* que contiene la herramienta *dnsmasq* para manejar los nodos de la red y llevar un registro de los hosts conocidos; *install_slurm.sh* instala y configura el servicio de *slurm* para añadir las ventajas que éste incorpora al sistema y que se han comentado en el apartado de herramientas; scripts como *locale.sh* y *network_lib.sh* se utilizan como librerías, ya que cuentan con funciones para instalar y definir el idioma, en el caso de *locale.sh*, o funciones para obtener las direcciones IP o las interfaces de la red en el caso de *network_lib.sh*, elementos que son necesarios para poder configurar el sistema de manera dinámica y general, para que cualquier persona que lo haga, en sus condiciones particulares, lo pueda llevar a cabo exitosamente; *add_monitoring.sh* instala y configura todo lo relacionado con la página web interna del clúster y el sistema de administración remota vía web, desarrollado por Joan Jara.

De esta manera, a modo visual para que queden más claras todas las interacciones que se dan cuando se configura por primera vez el nodo máster, se puede elaborar el siguiente diagrama con los scripts que se ejecutan y las acciones que se llevan a cabo durante este proceso de instalación.

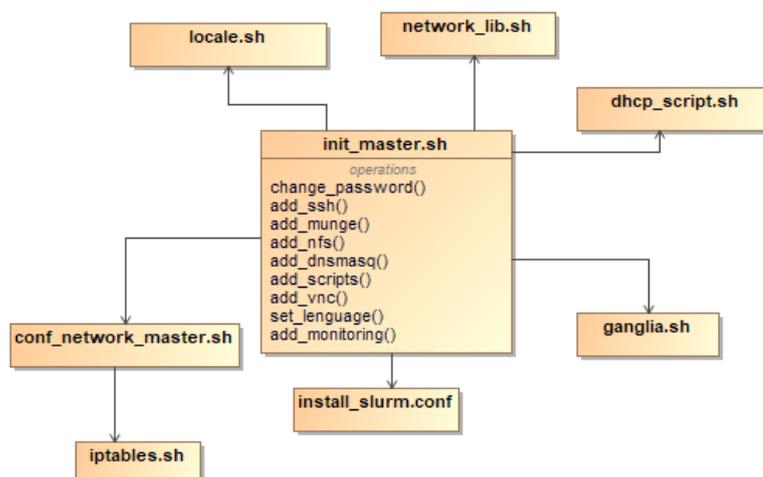


Figura 2. Diagrama de scripts de la inicialización del máster.

Por otro lado, el segundo subgrupo en el que podríamos dividir la arquitectura del software sería el relativo a la instalación y configuración de los nodos *slaves*. Este proceso lo desencadena el usuario administrador cada vez que quiere añadir un nuevo nodo al clúster. Para ello, ejecuta el script *add_slave.sh* con un argumento o parámetro de entrada, el cuál va a ser el nombre que se le ha asignado vía servidor DHCP a la nueva placa que se quiere añadir al sistema. Un ejemplo de este caso sería en un sistema que cuenta con una placa máster y 2 nodos *slaves* ya configurados, si se le quiere añadir un tercer nodo, el administrador debería ejecutar `./add_slave.sh odroid3`. Este script se encargará de establecer la conexión vía SSH con la nueva placa a configurar, pasará los scripts necesarios para ello, creará el usuario *odroid* en el nuevo nodo mediante el script *add_user.sh* (debido a que en la versión de sistema operativo escogida únicamente viene el usuario *root* por defecto, esto lo explico en detalle en la sección Decisiones de diseño) y ejecutará el script *init_slave.sh* en remoto en el nuevo nodo. Este script está diseñado de manera análoga al script *init_master.sh* y su objetivo es instalar y configurar todos los paquetes y herramientas necesarias de manera automatizada, para incorporar el nodo al clúster y poder ejecutar utilizando sus recursos y poder monitorizarlo. Este script se encarga de instalar la mayoría de paquetes que se instalaron en el máster, pero configurándolos para el nodo *slave*, como son el *slurm*, el *munge*, el *nfs* o el *ganglia*. Además, se encarga de configurar las interfaces de red y el DNS hacia el máster, de manera que utilice el nodo máster para sus comunicaciones, ya sea en la red interna o hacia internet.

De igual manera que con el máster, he elaborado un diagrama en el que se muestra la interacción entre los scripts y las acciones principales que se llevan a cabo, para mostrar de una forma más clara este proceso de configuración de un nuevo nodo *slave*.

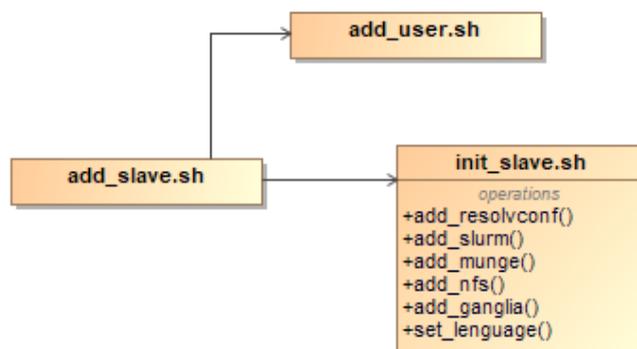


Figura 3. Diagrama de scripts de la inicialización de un nodo slave.

4.1.2 Diseño de la red

La arquitectura con la que está diseñada nuestra red ya se ha comentado y vendría a representar una típica LAN. En ella contamos con un nodo que ejerce la función de máster y a la vez sirve como firewall o cortafuegos de la red interna, y contamos con un conjunto de nodos, denominados slaves, que se encuentran dentro de esta red interna. De esta manera, una de las características principales de la red es que el nodo máster cuenta con 2 interfaces de red: una externa, conectada al router, para tener conexión con el exterior y acceso a internet; y otra interna, que le otorga conexión con todos los nodos de la red interna, formando así el clúster. Por lo tanto, otra característica principal de nuestro sistema es que los nodos slaves no tienen conexión directa con el exterior ni con internet, todos sus paquetes deberán pasar a través del nodo máster, el cual se encarga de filtrarlos y remitirlos a su destino, en el caso que esté permitido. Para llevar a cabo esa función de firewall, se utiliza `iptables`, la cual es una herramienta para la manipulación de las tablas que permiten el paso de paquetes. Esta herramienta es de tipo *packet-filtering stateful*, es decir, filtra paquetes pero considera el estado de las conexiones de la red, es decir, es capaz de identificar los paquetes que pertenecen a una comunicación que nosotros hayamos establecido o aceptado. Esta herramienta es muy útil, ya que se ha configurado con políticas restrictivas, es decir, que por defecto no deja pasar ningún tipo de paquete (lo cual es una buena práctica, ya que hace más seguro nuestro sistema) y a estas políticas se incorporan normas, para que aquellos paquetes que cumplan las condiciones especificadas en ellas, se les permita el paso. Estas normas vienen todas definidas en el script `iptables.sh` y, en líneas generales, incorporan reglas para permitir el paso de conexiones establecidas por nosotros, y por medio de puertos que utilizan las herramientas que hemos instalado y configurado en nuestro sistema, como es el puerto 22 del protocolo TCP para las conexiones vía SSH, el puerto 53 para conexiones del DNS, 68 para el servidor DHCP, 80 y 443 para HTTP y HTTPS, o los puertos que utilizan Ganglia, NFS o VNC, entre otros.

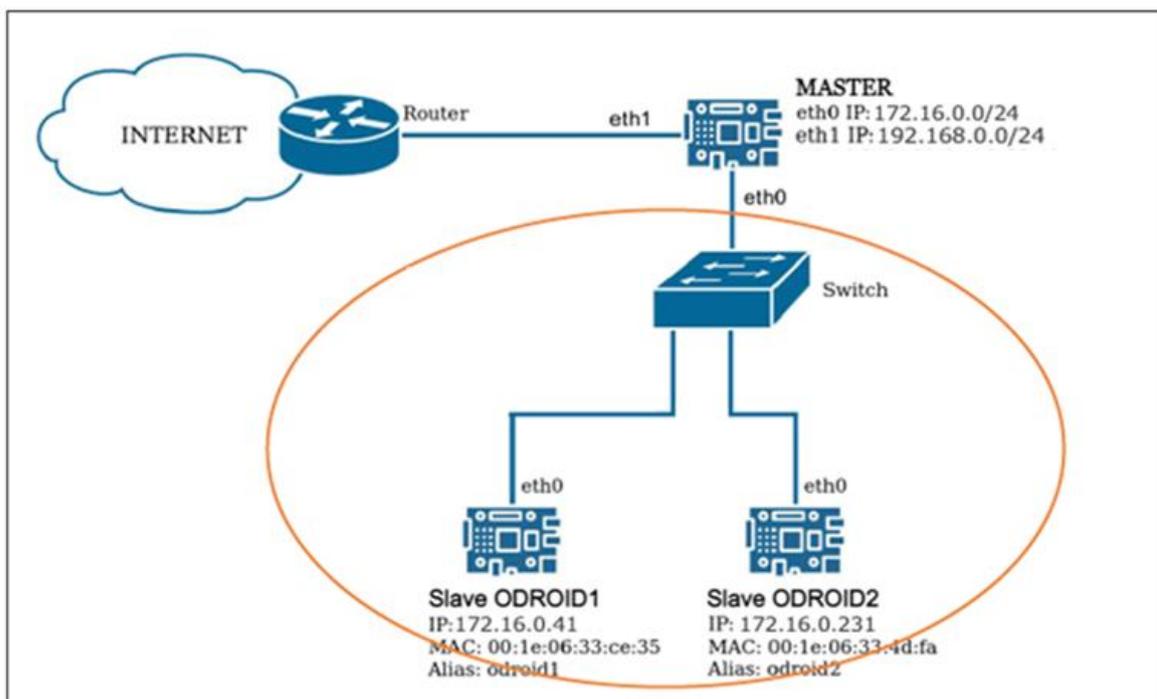


Figura 4. Diseño de la red en la que se ha elaborado este TFG.

4.2 Decisiones de diseño

En este apartado comento las decisiones de diseño más relevantes que he tenido que tomar para adaptar el proyecto a las características deseadas, manteniendo los elementos que se han podido, con modificaciones o no, y aportando otros necesarios para que el resultado final fuera funcional y lo más óptimo posible.

4.2.1 Para la configuración del máster

La primera de las decisiones que tomamos para la realización de este proyecto fue la versión del sistema operativo a utilizar. Esta decisión la tomamos en conjunto mi tutor Carles Aliagas y yo. La primera premisa de esta decisión es la de utilizar la versión de sistema operativo de Ubuntu más reciente a fecha de redactar este trabajo, ya que así tendríamos un sistema lo más actualizado posible. Por ello, decidimos utilizar la versión 20.04, mientras que mis predecesores habían realizado sus TFGs basados en la versión 18.04. Ubuntu dispone de 2 versiones dentro de Ubuntu 20.04: la versión Mate y la versión Minimal. Mate es una versión que incorpora por defecto más paquetes y herramientas, y por lo tanto es más pesada. Nos encontramos que la imagen de Ubuntu 20.04 Mate tiene un tamaño cercano a los 8 GB. Esto es un tamaño demasiado grande para el tipo de sistemas que queremos montar, ya que en general a las placas Odroid C2 se les pone una tarjeta de memoria eMMC que suele tener un tamaño de 8 GB. Por lo tanto, es inviable esta versión. Optamos, entonces, por utilizar la versión Ubuntu 20.04 Minimal. Esta versión es más ligera, con un tamaño de unos 3.8 GB y, por lo tanto, se adapta mejor a las condiciones del hardware. Además, un aspecto a tener en cuenta es que como nuestro software ya instala y configura todas las herramientas necesarias para el uso del sistema, no necesitamos todas las herramientas que incorpora la versión Mate. Debido a esto, mis predecesores en este proyecto debían eliminar todo aquello que no fuera necesario de la imagen, para liberar espacio y no tener un sistema casi saturado en cuanto a memoria. En mi caso, al utilizar la

versión Minimal, no he requerido de este paso y he eliminado de los scripts el código que realizaba esta desinstalación.

Fruto de utilizar la versión Minimal de Ubuntu 20.04, surge la primera incorporación que debemos realizar y es la creación de un segundo usuario para la imagen. La imagen Minimal, al contrario que Mate, tiene por defecto un único usuario activo: *root*. La decisión que se ha tomado es la de crear un nuevo usuario en todas las placas, tanto en la máster como en los slaves. Este usuario por defecto está configurado como usuario *odroid*, aunque esto se puede sustituir en el archivo *odroid_cluster.conf*, un archivo creado con algunas de las características básicas del sistema y que utilizan los scripts para configurarlo. Tener un usuario por defecto además del usuario *root* es una buena práctica, para no tener que ejecutar todo como administrador, y además es necesario para herramientas como slurm, que requiere que quien ejecute los comandos que esta herramienta proporciona sea un usuario no root y que además se encuentre en todos los nodos del clúster.

Este cambio que incorpora la versión Minimal en cuanto a los usuarios por defecto también ha originado un cambio a la hora de ejecutar los scripts de inicialización del sistema. Anteriormente, en las versiones de mis predecesores, los scripts se ejecutaban mediante el usuario *odroid*, utilizando la directiva *sudo*. Debido a que en el sistema por defecto este usuario no existe y siguiendo la coherencia de que este tipo de tareas solo sean llevadas a cabo por el administrador, se ha tomado la decisión de que los scripts de *init_master.sh* y *add_slave.sh* sean ejecutados por el usuario *root*.

Otra decisión que se ha tomado es la de incluir un navegador web junto con la instalación y configuración del máster. Al trabajar con la versión Minimal, al contrario que con la Mate, ésta no cuenta con ningún navegador web por defecto. Se ha decidido instalar Midori, ya que es un navegador ligero y que cumple con las necesidades de nuestro sistema. Es necesario un navegador, ya que las páginas de visualización de los gráficos de las herramientas de monitorización requieren que se visualicen desde la placa máster y no desde un ordenador externo a la red interna, ya que si no, las métricas de los nodos slave no se podrán visualizar.

4.2.2 Para la configuración del slave

Las decisiones de diseño tomadas para la configuración de los nodos slaves siguen la misma línea que las del máster y están orientadas a la creación del usuario no root. Las placas que ejercen la función de nodos slaves poseen la misma versión de sistema operativo que la máster, la Ubuntu 20.04 Minimal. Por lo tanto, la primera decisión que se ha tomado es la de crear también en éstas un usuario no root, pero con permisos de *sudo*, que por defecto se le ha llamado *odroid*. Éste es un cambio necesario para aspectos como la compartición de la clave pública RSA, que nos permitirá la conexión vía *ssh* a los nodos slave sin necesidad de autenticación por contraseña, el uso de las funciones de slurm, que requieren que exista el mismo usuario que ejecuta en la placa máster en los nodos slaves, o para la compartición del directorio *home* en el master del usuario *odroid* entre todos los nodos, ya que es una decisión que tomaron mis predecesores y que se ha decidido mantener, ya que es práctica y nos facilita la ejecución en el clúster.

Otro aspecto de primera relevancia que se ha establecido es la configuración de todos los nodos slave que se encuentren conectados a la red interna en el momento de la inicialización del máster de manera automática. Esto se realizará utilizando el sistema que ya estaba desarrollado, que es mediante el script *dhcp_script.sh* que se ejecuta desde el servidor DHCP de dnsmasq. Este script guarda las características de red de los nodos que

establecen conexión con el máster y ejecuta el script `add_slave.sh` para configurar el nodo slave de manera automática y transparente al usuario, desde la configuración inicial del sistema. De esta manera, obtenemos una configuración más automatizada y sencilla para el usuario.

En cuanto al resto de paquetes y herramientas, se ha mantenido el diseño llevado a cabo por David Ferrer en su trabajo, ya que se instala y configura lo necesario para que el sistema funcione tal y como se espera de él.

4.2.3 Herramienta de monitorización

Para la herramienta de monitorización principal del sistema, se barajaron diversos softwares de monitorización de clústeres y redes que existen actualmente para sistemas como el nuestro. Inicialmente, los candidatos fueron Monitorix, Ganglia y Cacti. Todos son sistemas de monitorización que permiten visualizar tanto las métricas y rendimiento de los nodos como el estado de la red. Sin embargo, tras el proceso de instalación, Cacti fue descartado debido a los problemas que ocasionó para su instalación y configuración, y debido a que este proceso es difícilmente automatizable, por lo que no nos interesa en este proyecto, lo que se busca es una herramienta ligera y de fácil configuración y uso.

De los dos candidatos que restaban, Ganglia y Monitorix, se ha realizado la instalación y prueba de ambos en el sistema. Esta evaluación viene detallada más en profundidad en el apartado de Evaluación, y consiste principalmente en instalar y configurar las herramientas de modo que recojan las métricas de los 2 nodos slave con los que cuento para desarrollar y probar este proyecto, además del master, y ejecutar una serie de scripts que utilizan OpenMPI para ver el rendimiento del sistema y si las herramientas de monitorización muestran diferencias en las lecturas, que pudieran ser ocasionadas por la propia herramienta en sí.

Tras esta evaluación, se ha decidido que la herramienta de monitorización que se queda de forma definitiva en el sistema sea la de Ganglia, por los siguientes motivos. El primero es que es una herramienta que cuenta con una fácil instalación y configuración, tanto en el nodo máster como en los slaves. Los ficheros de configuración de ésta en los distintos nodos son independientes, de manera que se configura por una parte el archivo del nodo máster, y cuando se han de añadir nodos slaves, únicamente hay que modificar el archivo del nodo slave. Esto ocurre en contraposición a Monitorix, la configuración del cual depende casi íntegramente del archivo de configuración del nodo máster, por lo que cada vez que se quiere añadir un nuevo nodo al sistema, se ha de modificar el archivo del máster. Esto, además de ser más complejo en cuanto a implementación de manera automatizada, conlleva el riesgo de que, al añadir un nuevo nodo, ocurra un error y desestabilice la configuración de todo el sistema. El segundo motivo por el que se ha optado finalmente por Ganglia como herramienta de monitorización, es porque esta herramienta cuenta con multitud de funcionalidades en su página de monitorización, como son la visualización de los gráficos, agregarlos para ofrecer una vista distinta, comparar hosts, programar eventos, etc. Monitorix, por otro lado, es una herramienta que está todavía en desarrollo y únicamente dispone de la posibilidad de visualizar los gráficos. Por este motivo, considero que es más interesante tener una herramienta con más funcionalidades que no solo visualizar, ya que será de más uso sobre todo si hablamos de entornos docentes, que es para lo que está pensado este proyecto, ya que permitirá al administrador/profesor realizar aspectos interesantes con los datos.

4.2.4 Herramientas de administración remota

Para las tareas de administración remota del clúster, se ha optado por mantener la herramienta de mantenimiento que implementó Joan Jara en su trabajo. Ésta está disponible en la página web interna del nodo máster y proporciona una forma de ejecutar comandos en remoto en los nodos seleccionados mediante una sencilla interfaz gráfica.

Sin embargo, debido a que esta herramienta tiene limitaciones, como es que no se pueden ejecutar comandos que requieran de permisos de administrador o sudo, también se ha decidido incluir unos scripts que ejecuten en remoto y que sí que permitan llevar a cabo estas tareas de administración y mantenimiento. Estos scripts se proporcionan junto con el software en una carpeta llamada *bin* y el proceso de instalación y configuración del máster los coloca bajo el directorio *home* del usuario por defecto, de modo que este usuario podrá ejecutarlos.

5 Implementación

5.1 Cambios en los scripts para la configuración del máster

La primera parte de mi trabajo ha sido la de revisar todo lo realizado por David Ferrer en su TFG, predecesor del mío, y realizar los cambios necesarios para adaptar los scripts a la última versión de Ubuntu en el momento de realizar este trabajo. El primer cambio respecto a lo hecho hasta ahora, como se ha comentado, es la versión del sistema operativo de las placas. Se ha pasado de una versión Mate, que como se ha dicho incorpora por defecto muchas herramientas y paquetes, a una versión Minimal, que tiene lo justo y necesario para que se ejecute la imagen. Es por ello que, durante el proceso de instalación y configuración, han surgido muchos problemas derivados por la falta de dependencias y actualizaciones de paquetes que no se instalaban o configuraban de la misma manera.

5.1.1 *init_master.sh*

A continuación se detallarán los cambios que se han realizado sobre el script *init_master.sh*, el cual es el script que instala y configura el nodo máster del clúster.

Esta versión, a diferencia de la Mate, que viene con 2 usuarios (*root* y *odroid*), solo viene por defecto con el usuario *root*. Es por ello por lo que uno de los cambios que se realiza sobre el script *init_master.sh* es añadir un nuevo usuario al sistema, *odroid* por defecto, se le asigna una contraseña distinta de la del usuario *root*, que introduce el administrador a la hora de ejecutar el script y se le otorgan permisos para ejecutar *sudo*. Esto se realiza utilizando el comando `adduser --gecos "" odroid`, se escribe la nueva contraseña, y a continuación se añade este usuario al fichero *sudoers* para que tenga acceso a *sudo*.

Un cambio que se ha realizado al inicio de este script viene dado por la situación de que en la anterior versión 18.04 de Ubuntu, había un error por una clave de los servidores de Ubuntu que no estaba actualizada y se debía hacer antes de poder actualizar e instalar nuevos paquetes. Este problema ya está solucionado en la versión 20.04, por lo que el código que añade esta clave ya no es necesario y se ha eliminado.

Como se ha comentado, la versión de Ubuntu que se ha escogido para que sea la definitiva pesa mucho menos que la Mate, debido a que incorpora muchos menos paquetes y funcionalidades. Debido a ello, hay muchas líneas en el script *init_master.sh* que ya no son necesarias. Hablamos de código que se encarga de desactivar los intentos de actualización automáticos, al inicio del script, ya que esta versión no tiene esta funcionalidad; o líneas que se encargaban de eliminar software innecesario para nuestro clúster, ya que esta versión más reducida no contiene ese software de base. Todo este código se ha eliminado ya que, pese a que no suponía una modificación del funcionamiento del sistema, sí que suponía un aumento en el tiempo de ejecución del script *init_master.sh*, además de indicar por el terminal una serie de errores que es preferible evitar de cara a que el usuario ajeno a este proyecto que ejecute el script no vea esos errores que podrían hacerle pensar que algo va mal en el proceso de instalación.

Durante la instalación del servicio *chrony*, que se utiliza para tener la hora actualizada, se ha detectado un problema que impedía que este servicio fuese activado. El problema consistía en que no podía cargar las reglas SECCOMP. Éstas son parte de una política referida a la computación segura que se compila con el kernel de Linux. No sé si es debido a que en esta versión de Ubuntu Minimal no están incluidas o si se debe a otro problema. Este error resulta clave, ya que al instalar *chrony*, el servicio para actualizar la hora que tiene *systemd*, *systemd-timesyncd* se desinstala para que no cree conflictos, y al

no poder iniciar *chronyd*, el *daemon* de *chrony*, el sistema se queda sin servicio de actualización de hora, lo cual afecta a otros procesos como las herramientas de monitorización. Es por ello que se ha decidido no instalar la herramienta *chrony* y mantener *el systemd-timesyncd* que viene por defecto en la imagen.

Otro cambio que es notable destacar se encuentra en la función *add_dnsmasq*, del script *init_master.sh*. Esta función tiene como objetivo instalar y configurar la herramienta *dnsmasq* para que sirva, además de como DNS, como servidor DHCP para la red interna. El problema con esta función es que, pese a funcionar correctamente, arrojaba un error inicial en la configuración de *dnsmasq*, lo cual puede confundir al usuario que instala el sistema y ve cómo aparece en la línea de comandos errores y fallos. Como el error solo se producía la primera vez que se ejecutaba, y al finalizar la instalación ya se encontraba funcionando correctamente el servicio de *dnsmasq*, se ha optado por redirigir la salida de errores para que esta sea transparente al usuario. Sin embargo, tras varias pruebas no se ha conseguido redirigir el error para que no se muestre. Aun así, dentro de que el sistema de instalación es largo y arroja mucho contenido por el terminal, el usuario no debería tener problema a la hora de ejecutarlo, ya que no afecta para nada en el funcionamiento del sistema y al finalizar la ejecución podrá comprobar cómo el servicio de *dnsmasq* funciona correctamente.

En cuanto a paquetes que por defecto estaban incorporados en la versión Mate y que por otro lado en ésta no se encuentran, se ha añadido código para instalar algunas herramientas con las que viene bien contar o que eran requeridas por otros scripts, como son el paquete *ifupdown*, cuyas funciones *ifdown* e *ifup* utiliza el código de la función *add_dnsmasq*, o el paquete *net-tools*, el cual incorpora herramientas para la visualización y manejo de la red como el comando *arp* para ver los dispositivos conectados a ambas interfaces del nodo master.

5.1.2 *install_slurm.sh*

En el proceso de instalación, advertí que el servicio de *slurm* no se configuraba correctamente y no se podía iniciar. Me di cuenta que uno de los archivos de configuración del *slurm*, el *cgroup.release_common*, no se descargaba durante el proceso de instalación del paquete *slurm-wlm*, por lo que no se podía copiar a */etc/slurm-llnl/cgroup* y no se configuraba correctamente. Para solucionar esto, opté por incluir este archivo en el zip que contiene todos los scripts para la configuración del sistema y copiarlo desde allí. Una vez más, este problema puede haber sido originado porque la versión de Ubuntu Minimal no cuenta con este archivo, aunque de esta manera nos aseguramos de que no falle el proceso de configuración de *slurm* en cualquier sistema porque siempre contará con el archivo. El segundo cambio que he tenido que realizar para que se inicien correctamente los servicios de *slurm* es modificar el script *install_slurm.sh*. En esta nueva versión de Ubuntu, el sistema aloja los PID de los servicios bajo el directorio */run/*, y no bajo */var/run/*, como lo hacía anteriormente. Es por ello que en el fichero de configuración */etc/slurm-llnl/cgroup.conf*, he tenido que modificar el path a los PID tanto del servicio *systemd* como de *systemd* a */run/systemd.pid* y */run/systemd.pid*, respectivamente.

5.1.3 *add_monitoring.sh*

Por último, se ha suprimido la función *add_monitoring* que instalaba el software de Joan Jara para configurar la página web interna y el sistema de administración remota y se ha movido a un script por separado, llamado *add_monitoring.sh*. Este cambio es debido a que facilita la ejecución del script, que se llama desde el script *init_master.sh*. El código que antes comprendía esta función se ha modificado en varios aspectos: el primero de ellos

es que la descarga del software de Joan Jara se realizaba mediante el comando *git clone*. Esta decisión me parece desacertada ya que primero depende de que Joan Jara mantenga en activo su página de Github y no modifique el contenido, y segundo que esto no permite la modificación del código de su software por otro que no sea él, algo que de no haberse realizado no hubiera sido posible que se instalara y configurara correctamente su software debido a problemas de dependencias que más adelante explicaré. La implementación final cuenta con el software de Joan incorporado dentro del archivo comprimido general, de manera que facilita su manipulación y no depende de páginas externas como Github. Además de la forma en la que se incorporaba el software, también se ha modificado el directorio en el que se ubicaba este software, que era bajo el directorio ‘Downloads’ del usuario odroid. Me ha parecido más conveniente crear un directorio específico para éste y así se ha hecho.

En cuanto a los problemas de dependencias que originó la implementación del software de Joan en el sistema, principalmente eran debido a falta de paquetes, al igual que casi la totalidad de problemas de dependencias que ha habido en el proyecto, por el uso de una versión con menos paquetes de fábrica. Se han instalado paquetes como *testresources* o *python3-six* para solucionar estos problemas de dependencias en el archivo *start_monitoring.sh*. Otro problema que surgía con la instalación de este software venía dado en el archivo *pm2.sh*. Este script se encarga de configurar el servidor para Node.js, el cual utiliza Joan para su página web y sistema de monitorización. En este caso, el problema de dependencias ocurría porque no existía el archivo *package.json* bajo del directorio *.django-monitor* bajo del directorio *home* del usuario. Para solucionar este problema, primero se utilizó una sentencia de código que lo crea. Sin embargo, esta ejecución es interactiva y si establecemos la no interactividad y dejamos los parámetros por defecto, no se ejecuta correctamente. La implementación final se realiza en el script *pm2.sh* y consiste en un echo del contenido que debe llevar el script correctamente configurado redirigido para que cree el archivo *package.json*. Este método para establecer archivos de configuración ha sido ampliamente utilizado en este proyecto ya que facilita y automatiza estos procesos con el contenido exacto que queremos.

5.1.4 Administración remota vía scripts

La última modificación a este script de configuración del máster, el *init_master.sh*, es la incorporación de una función, llamada *add_scripts*, que se encarga de colocar, bajo el directorio *home* del usuario, los scripts para la administración remota que se han comentado en el apartado 4.2.4 y cambiar los permisos de éstos para que puedan ser ejecutados tanto por el administrador como por el usuario por defecto odroid. La implementación de estos scripts se basa en el comando *ssh* con la opción *-t*, que permite ejecutar un script o comando en remoto. Estos scripts consisten en un bucle que itera por todos los nodos del clúster, leyendo esta información de un archivo, llamado *odroid.par*, que se crea con la adición del primer nodo slave al sistema y se ubica bajo el directorio */etc* y que incrementa su contenido a medida que se añaden nodos slave al clúster. De esta manera, gracias a este archivo, si se quiere excluir alguno de los nodos del clúster de una ejecución (ya sea porque se ha colgado y no responde o por cualquier otro motivo) se puede fácilmente comentar o eliminar la línea de este archivo y no se tendrá en cuenta para la ejecución.

Se incorporan 2 scripts para realizar esta función de administración remota, los cuales tienen un código idéntico excepto por el uso del símbolo ‘&’ en el comando *ssh* en uno de ellos. Lo que nos permite esto es que, en el script que no cuenta con el símbolo ‘&’, el comando *ssh -t* seguido del comando o script a ejecutar en todos los nodos se realice

de manera secuencial, ya que tendrá que esperar a la finalización de una ejecución para mandar la siguiente. Esto sobre todo lo usaremos para tareas de actualización y de cualquiera que implique un paso de paquetes por el nodo máster, ya que como éste ejerce de cortafuegos de toda la red interna, ejecutar estos comandos a la vez en todos los nodos provocaría una saturación del nodo máster.

En cambio, en el script que cuenta con el símbolo ‘&’, la ejecución continúa sin esperar a que termine el comando, por lo que se lanza a todos los nodos de la lista, a todos los nodos del clúster. Para ejecutar comandos o scripts que no requieran de paso de paquetes, como por ejemplo los comandos *reboot* o *halt*, este script es óptimo ya que se ejecutará en todos a la vez sin demora.

5.2 Cambios en los scripts para la configuración de los nodos slaves

Los nodos catalogados como slaves tienen la misma versión que el máster, la Ubuntu Minimal 20.04, por lo que igualmente se han tenido que realizar cambios en los scripts *init_slave.sh* y *add_slave.sh* para su correcta instalación y configuración.

5.2.1 *add_slave.sh*

Primero de todo trataremos los cambios en el script *add_slave.sh*, el cual se encarga de establecer la conexión con el nodo que se quiere añadir al clúster, crear el usuario odroid, copiar los archivos necesarios y ejecutar el script *init_slave.sh* para configurar el sistema automáticamente.

Lo primero que cabe destacar es que se ha modificado el script para permitir que se ejecute desde el script *dhcp_script.sh*, de manera que la primera vez que un nodo conectado a la red establezca conexión con el máster, el servidor DHCP que lleva la herramienta dnsmasq ejecutará este script y en él se llama al script *add_slave.sh* para inicializar y configurar el nodo slave. Para que esto sea posible, se realizan cambios en este script que comento en los siguientes párrafos.

Debido a que la versión Minimal de Ubuntu 20.04 únicamente viene con un usuario de base, el usuario root, debemos crear el usuario odroid para ejecutar, entre otros, el slurm y tener la configuración del NFS y del munge bien adaptada. Para añadir este usuario, se ha optado por hacerlo de una manera diferente a la que se crea el usuario odroid en el máster. En este caso, con el objetivo de que las contraseñas del usuario odroid de todos los nodos sean la misma, en vez de utilizar el comando `adduser`, lo que se hace es copiar las líneas correspondientes al usuario odroid de los ficheros */etc/passwd*, */etc/group*, */etc/shadow* y */etc/shadow-* del master al slave, se crea el directorio *home* del usuario odroid en el slave, que posteriormente será compartido mediante NFS y se añade al archivo *sudoers* para otorgarle permisos de sudo. De esta manera, el usuario odroid del slave tendrá la misma contraseña que el del master y compartirán el mismo directorio home. Durante la realización de esta parte de código, me encontré con un problema a la hora de copiar la contraseña del fichero */etc/shadow*. Ésta puede contener símbolos, ya que se encuentra encriptada, como el ‘\$’ o el ‘&’ que tienen un significado o realizan una función en la Shell. A consecuencia de esto, al copiar la contraseña, el intérprete de comandos las interpreta con su función y esto da lugar a que el fichero */etc/shadow* del nodo slave no contiene la misma contraseña que el del master. Para solucionar esto, se optó por crear un script aparte para realizar la adición del usuario odroid, llamado *add_user.sh*, y se trata la cadena correspondiente a la contraseña añadiendo comillas simples para que la interprete como únicamente texto y no con la función de estos símbolos.

Junto a estos cambios, el script estaba adaptado para establecer la conexión ssh al usuario odroid del slave, pero como ya hemos comentado, éste no se encuentra por defecto en la versión Minimal de Ubuntu. Es por ello que se ha decidido cambiar esto para que la conexión se haga al usuario root de los nodos slave y la inicialización de éste se haga como usuario root. Para ello, lo que primero se realiza es una compartición de la clave pública del usuario root para que las posteriores conexiones no requieran de una contraseña vía terminal, algo que, debido a que el script en general se ejecuta desde el daemon del dnsmasq mediante el *dhcp_script.sh*, esto no permite la interacción como el usuario. Compartir la clave pública de root con los slaves es una ventaja que nos permitirá posteriormente ejecutar los scripts de administración remota, el cuál necesita de permisos de root en la mayoría de los comandos de mantenimiento y administración, ya que estos scripts establecen conexión vía SSH con los nodos slave y de esta manera no se pedirá la contraseña de root de los slaves en cada ejecución. De esta manera, las modificaciones en el script *add_slave.sh* se traducen en modificar las conexiones de *sshpass*, *scp* y *ssh* para que lo hagan al usuario root del nodo slave y así se pueda configurar éste automáticamente desde la ejecución del *dhcp_script.sh* de manera transparente al usuario.

Para finalizar con el apartado correspondiente al script *add_slave.sh*, quería comentar que se han añadido algunos mensajes para indicar al usuario ciertos procesos que deberá realizar, como introducir contraseñas o verificar que la inicialización del nodo slave ha iniciado, que tardará un tiempo, y cuándo haya finalizado correctamente.

5.2.2 *init_slave.sh*

En el script *init_slave.sh*, de manera análoga que en el *init_master.sh*, se ha eliminado código sobrante que no tenía función en esta nueva versión, como es el código que actualizaba la clave de los servidores de Ubuntu por un error de la antigua versión o como es el código que desinstalaba una multitud de paquetes innecesarios para la configuración del clúster, que en la versión Minimal no viene incluida y por lo tanto no es necesario. Al inicio del script se han añadido dos líneas de código con el objetivo de solucionar un error que originaba el idioma. Estas dos líneas son: *locale-gen es_ES.UTF-8* y *dpkg-reconfigure --frontend noninteractive locales*. El problema no estoy seguro si era de dependencias o por qué se originaba, ya que en el nodo máster no se daba y tienen la misma versión de sistema operativo. Sin embargo, con esta adición se soluciona.

En el script he eliminado, al igual que en el máster, el código que instalaba el servicio *chrony* por los mismos motivos que en el máster. Además de esto, he añadido una función en el *init_slave.sh*, llamada *add_ganglia*, que se encarga de instalar y configurar el software necesario para contar con la herramienta de monitorización en el slave y que ésta envíe sus métricas al nodo máster.

Por último, cabe comentar que he tenido que comentar la línea que modificaba el archivo */etc/ssh/sshd_config* para no permitir el login a los slaves como root, ya que los scripts añadidos para la administración remota utilizan *ssh* para conectarse a los nodos slave y deben hacerlo como root para tener permisos para ejecutar comandos de administrador.

5.3 Scripts para las herramientas de monitorización

En esta sección comentaré la implementación de los scripts que permiten la instalación y configuración de las herramientas de monitorización que he probado, Monitorix y Ganglia.

El primer aspecto que cabe comentar es la adición de nuevas reglas al fichero *iptables.sh* que permiten el paso de mensajes de las herramientas de monitorización entre los nodos y el máster y que nos permiten tener acceso a sus correspondientes páginas web dónde podemos visualizar las métricas recogidas. Por defecto, Monitorix utiliza el puerto 8080 tanto para el paso de mensajes entre nodos como el propio servidor web que incorpora para la visualización de los gráficos. Utiliza el protocolo TCP, por lo que estas reglas han sido añadidas a la tabla de iptables. Por su parte, Ganglia se comunica por defecto mediante el puerto 8649. Utiliza 2 protocolos, TCP para acceder a la página web, ubicada mediante servidor Apache, por lo que hay que permitir el paso de paquetes por el puerto 80, mientras que para el paso de mensajes entre nodos utiliza el protocolo UDP. Estas reglas han sido añadidas al script *iptables.sh* para que puedan visualizarse las métricas de todos los nodos.

5.3.1 Monitorix

Para configurar y probar Monitorix, he implementado un script para instalarlo y configurar el máster, llamado *monitorix.sh*, y otro para instalarlo en los nodos slave, llamado *monitorix_slave.sh*. Estos scripts básicamente realizan el proceso de instalación y proporcionan una configuración mínima. Sin embargo, para poder visualizar los datos de los nodos slaves, se tiene que modificar el archivo *monitorix.conf* del máster para incluir las direcciones IP de los nodos slaves. Este proceso no se ha incluido en los scripts ya que, como se ha comentado en el apartado de diseño, finalmente se ha escogido Ganglia para la versión definitiva del sistema. Por lo tanto, no son necesarios estos scripts en la versión final y las pruebas se han realizado configurando manualmente, editando el fichero de configuración de monitorix con las IP de las dos placas slave que he utilizado para testear el sistema.

Con la herramienta Monitorix, al igual que con Ganglia, me surgió un problema inicialmente a la hora de visualizar las métricas de los nodos slave, las cuales no llegaban a verse. El problema no era de la configuración de Monitorix o del sistema, sino que como yo me estaba conectando a la página de Monitorix desde mi ordenador, el cual no se encuentra dentro de la LAN interna, esos datos que enviaban los nodos slave yo no podía visualizarlos. Es por este motivo por el cual se optó por incluir un navegador en la instalación del sistema, como se ha comentado en el apartado de decisiones de diseño, Midori, y es a través del comando `ssh` con la opción `-Y`, la cual permite el paso de X11¹⁴ y tras ejecutar el navegador Midori desde el máster, ya se pueden visualizar las métricas de los nodos slaves y constatar que la instalación y configuración de tanto Monitorix como de Ganglia se ha hecho correctamente.

5.3.2 Ganglia

Para configurar y probar la herramienta de Ganglia, he implementado un script, llamado *ganglia.sh*, que es llamado desde el script *init_master.sh* para configurar tanto el paquete *ganglia-monitor* y *ganglia-webfrontend*, así como el servidor Apache y el resto de

¹⁴ X11: es un sistema de visualización gráfico utilizado en sistemas Unix. Gracias a la opción `-Y` del comando `ssh`, que permite el paso de entornos gráficos, podemos visualizar el navegador Midori como si estuviésemos conectados desde la propia placa máster y no desde un PC externo a la LAN.

dependencias que éstos tienen. La herramienta *ganglia-monitor* se utiliza para monitorear el sistema en el que se instala, por lo que se hará tanto en el nodo máster como en los nodos slaves, mientras que la herramienta *ganglia-webfrontend* únicamente se instalará en el máster. Cada una de estas herramientas tiene un daemon o servicio asociado: la primera es *ganglia-monitor*, que como hemos comentado se encargará de recoger las métricas del sistema, y por su parte la segunda, *gmetad*, se encarga de recopilar estos datos y mostrarlos a través de la página web que utiliza Ganglia para la visualización de gráficos accesible mediante navegador web introduciendo en la barra de búsquedas la dirección `http://masterIP/ganglia`. El script *ganglia.sh* cuenta con el código para instalar los paquetes mencionados y realiza una pequeña configuración de los archivos de Apache para establecer algunos valores como el límite de la memoria o el tamaño máximo de los datos, así como establecer la zona horaria. Así mismo, también se modifica el contenido de los archivos de configuración tanto de *gmetad* como de *ganglia-monitor*, llamado *gmond.conf* para establecer las direcciones del nodo máster y así poder enviar los paquetes.

Por otro lado, para la configuración de Ganglia en los nodos slaves, como ya se ha comentado, el proceso de instalación es el mismo a excepción de que no se instala el paquete *ganglia-webfrontend*. La dirección IP a la que se referencia en los nodos slaves debe ser la misma que se establece en el archivo *gmetad.conf* del máster. Esto cabe destacarlo ya que el máster posee 2 direcciones IP, una para la red interna y otra para la externa, por lo que cabe recalcar que todas las IP en los archivos de configuración deben coincidir.

6 Evaluación

6.1 Evaluación de la configuración y automatización del sistema

Aquí evaluamos la correcta instalación y configuración tanto del nodo que hará la función de máster, mediante el script *init_master.sh*, como de los nodos slave, mediante el script *add_slave.sh*, con el objetivo de obtener un sistema paralelo distribuido plenamente funcional.

6.1.1 Evaluación del máster

A continuación se muestra la salida correspondiente a cada una de las principales acciones que se realizan para la inicialización del máster mediante el script *init_master.sh*. En algunos casos se ha acortado la salida para que no ocupe un número de páginas muy elevado.

6.1.1.1 Cambiar la contraseña de root y añadir usuario odroid

```
Enter the new password for the root user in master node:
```

```
Re-enter the new password:
```

```
Adding user `odroid' ...
```

```
Adding new group `odroid' (1000) ...
```

```
Adding new user `odroid' (1000) with group `odroid' ...
```

```
Creating home directory `/home/odroid' ...
```

```
Copying files from `/etc/skel' ...
```

```
New password:
```

```
Retype new password:
```

```
passwd: password updated successfully
```

6.1.1.2 Update

```
Unpacking alsa-topology-conf (1.2.2-1) ...
```

```
Selecting previously unselected package alsa-ucm-conf.
```

```
Preparing to unpack .../115-alsa-ucm-conf_1.2.2-1ubuntu0.8_all.deb ...
```

```
Unpacking alsa-ucm-conf (1.2.2-1ubuntu0.8) ...
```

```
Selecting previously unselected package apg.
```

```
Preparing to unpack .../116-apg_2.2.3.dfsg.1-5_arm64.deb ...
```

```
Unpacking apg (2.2.3.dfsg.1-5) ...
```

```
Selecting previously unselected package python3-httplib2.
```

```
Preparing to unpack .../117-python3-httplib2_0.14.0-1ubuntu1_all.deb ...
```

```
Unpacking python3-httplib2 (0.14.0-1ubuntu1) ...
```

```
Selecting previously unselected package python3-problem-report.
```

```
Preparing to unpack .../118-python3-problem-report_2.20.11-0ubuntu27.18_all.deb ...
```

```
Unpacking python3-problem-report (2.20.11-0ubuntu27.18) ...
```

```
Selecting previously unselected package python3-requests-unixsocket.
```

```
Preparing to unpack .../119-python3-requests-unixsocket_0.2.0-2_all.deb ...
```

```
Unpacking python3-requests-unixsocket (0.2.0-2) ...
```

```
Selecting previously unselected package python3-entrypoints.
```

```
Preparing to unpack .../120-python3-entrypoints_0.3-2ubuntu1_all.deb ...
```

```

Unpacking python3-entrypoints (0.3-2ubuntu1) ...
Selecting previously unselected package python3-cryptography.
Preparing to unpack .../121-python3-cryptography_2.8-3ubuntu0.1_arm64.deb ...
Unpacking python3-cryptography (2.8-3ubuntu0.1) ...
Selecting previously unselected package python3-secretstorage.
Preparing to unpack .../122-python3-secretstorage_2.3.1-2ubuntu1_all.deb ...
Unpacking python3-secretstorage (2.3.1-2ubuntu1) ...

```

Progress: [12%] [#####.....]

•
•
•

Created symlink /etc/systemd/system/dbus-org.bluez.service → /lib/systemd/system/bluetooth.service.

Created symlink /etc/systemd/system/bluetooth.target/wants/bluetooth.service → /lib/systemd/system/bluetooth.service.

```

Setting up libfontenc1:arm64 (1:1.1.4-0ubuntu1) ...
Setting up libclutter-1.0-common (1.26.4+dfsg-1) ...
Setting up libuv1:arm64 (1.34.2-1ubuntu1.3) ...
Setting up libllvm11:arm64 (1:11.0.0-2~ubuntu20.04.1) ...
Setting up libgles2:arm64 (1.3.2-1~ubuntu0.20.04.1) ...
Setting up emacsen-common (3.0.4) ...
Setting up libatasmart4:arm64 (0.19-5) ...
Setting up libblockdev-utils2:arm64 (2.23-2ubuntu3) ...
Setting up libepoxy0:arm64 (1.5.4-1) ...
Setting up libnspr4:arm64 (2:4.25-1) ...
Setting up ssl-cert (1.0.39) ...
Setting up libxfixes3:arm64 (1:5.0.3-2) ...
Setting up libxcb-sync1:arm64 (1.14-2) ...
Setting up libavahi-common-data:arm64 (0.7-4ubuntu7.1) ...
Setting up python3-wadllib (1.3.3-3build1) ...
Setting up wamerican (2018.04.16-1) ...
Setting up xfonts-encodings (1:1.0.5-0ubuntu1) ...
Setting up zenity-common (3.32.0-5) ...
Setting up libopus0:arm64 (1.3.1-0ubuntu1) ...

```

Progress: [61%] [#####.....]

•
•
•

```

Processing triggers for rygel (0.38.3-1ubuntu1) ...
Processing triggers for sgml-base (1.29.1) ...

```

6.1.1.3 Instalar net-tools

The following NEW packages will be installed:

net-tools

0 upgraded, 1 newly installed, 0 to remove and 94 not upgraded.

Need to get 194 kB of archives.

After this operation, 827 kB of additional disk space will be used.

Get:1 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 net-tools arm64 1.60+git20180626.aebd88e-1ubuntu1 [194 kB]

Fetches 194 kB in 1s (271 kB/s)

Selecting previously unselected package net-tools.

(Reading database ... 68510 files and directories currently installed.)

Preparing to unpack .../net-tools_1.60+git20180626.aebd88e-1ubuntu1_arm64.deb ...

Unpacking net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...

Setting up net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...

Processing triggers for man-db (2.9.1-1) ...

6.1.1.4 Instalar ifupdown

Suggested packages:

rdnssd

The following NEW packages will be installed:

ifupdown

0 upgraded, 1 newly installed, 0 to remove and 94 not upgraded.

Need to get 58.3 kB of archives.

After this operation, 233 kB of additional disk space will be used.

Get:1 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 ifupdown arm64 0.8.35ubuntu1 [58.3 kB]

Fetches 58.3 kB in 1s (111 kB/s)

Selecting previously unselected package ifupdown.

(Reading database ... 68579 files and directories currently installed.)

Preparing to unpack .../ifupdown_0.8.35ubuntu1_arm64.deb ...

Unpacking ifupdown (0.8.35ubuntu1) ...

Setting up ifupdown (0.8.35ubuntu1) ...

Created symlink /etc/systemd/system/multi-user.target.wants/networking.service → /lib/systemd/system/networking.service.

Created symlink /etc/systemd/system/network-online.target.wants/networking.service → /lib/systemd/system/networking.service.

Processing triggers for man-db (2.9.1-1) ...

Processing triggers for systemd (245.4-4ubuntu3.4) ...

6.1.1.5 Instalar Xfce

Unpacking xarchiver (1:0.5.4.14-1) ...

Selecting previously unselected package xdg-utils.

Preparing to unpack .../190-xdg-utils_1.1.3-2ubuntu1.20.04.2_all.deb ...

Unpacking xdg-utils (1.1.3-2ubuntu1.20.04.2) ...

```

Selecting previously unselected package xfburn.
Preparing to unpack .../191-xfburn_0.6.2-1_arm64.deb ...
Unpacking xfburn (0.6.2-1) ...
Selecting previously unselected package xfce4-appfinder.
Preparing to unpack .../192-xfce4-appfinder_4.14.0-1_arm64.deb ...
Unpacking xfce4-appfinder (4.14.0-1) ...
Selecting previously unselected package xfce4-panel.
Preparing to unpack .../193-xfce4-panel_4.14.3-1_arm64.deb ...
Unpacking xfce4-panel (4.14.3-1) ...
Selecting previously unselected package xfce4-pulseaudio-plugin:arm64.
Preparing to unpack .../194-xfce4-pulseaudio-plugin_0.4.3-0ubuntu1_arm64.deb ...
Unpacking xfce4-pulseaudio-plugin:arm64 (0.4.3-0ubuntu1) ...
Selecting previously unselected package xiccd.
Preparing to unpack .../232-xiccd_0.3.0-1_arm64.deb ...
Unpacking xiccd (0.3.0-1) ...
Selecting previously unselected package lm-sensors.
Preparing to unpack .../239-lm-sensors_1%3a3.6.0-2ubuntu1_arm64.deb ...
Unpacking lm-sensors (1:3.6.0-2ubuntu1) ...
.
.
.
Setting up cups-ipp-utils (2.3.1-9ubuntu1.1) ...
Setting up elementary-xfce-icon-theme (0.15-1) ...
Setting up gtk2-engines-murrine:arm64 (0.98.2-3) ...
Setting up unity-gtk-module-common (0.0.0+18.04.20171202-0ubuntu2) ...
Setting up libpaper1:arm64 (1.1.28) ...
Creating config file /etc/papersize with new version
Setting up libtie-ixhash-perl (1.23-2) ...
Setting up xfdesktop4-data (4.14.2-1) ...
Setting up liblouisutdml-data (2.8.0-3) ...
Setting up psmisc (23.3-1) ...
Setting up libxpresent1:arm64 (1.0.0-2build1) ...
Setting up libtagc0:arm64 (1.11.1+dfsg.1-0.3ubuntu2) ...
Setting up libcdio18:arm64 (2.0.0-2) ...
Setting up libencode-locale-perl (1.05-1) ...
Setting up libfcitx-utils0:arm64 (1:4.2.9.7-3) ...
Setting up fonts-quicksand (0.2016-2) ...
Setting up desktop-base (10.0.3ubuntu1) ...
update-alternatives: using /usr/share/desktop-base/debian-logos to provide /usr/share/images/vendor-logos
(vendor-logos) in auto mode
update-alternatives: using /usr/share/desktop-base/futureprototype-theme to provide /usr/share/desktop-
base/active-theme (desktop-theme) in auto mode
update-alternatives: using /usr/share/desktop-base/active-theme/wallpaper/contents/images/1920x1080.svg to
provide /usr/share/images/desktop-base/desktop-background (desktop-background) in auto mode

```

```

update-alternatives: using /usr/share/desktop-base/active-theme/wallpaper/gnome-background.xml to provide
/usr/share/images/desktop-base/desktop-background.xml (desktop-background.xml) in auto mode

update-alternatives: using /usr/share/desktop-base/active-theme/lockscreen/gnome-background.xml to provide
/usr/share/images/desktop-base/desktop-lockscreen.xml (desktop-lockscreen.xml) in auto mode

update-alternatives: using /usr/share/desktop-base/active-theme/wallpaper to provide
/usr/share/wallpapers/DebianTheme (desktop-plasma5-wallpaper) in auto mode

update-alternatives: using /usr/share/desktop-base/active-theme/login/background.svg to provide
/usr/share/images/desktop-base/login-background.svg (desktop-login-background) in auto mode

Setting up liblouis-data (3.12.0-3) ...
Setting up libdata-dump-perl (1.23-1) ...
Setting up libjbig2dec0:arm64 (0.18-1ubuntu1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/cups.path → /lib/systemd/system/cups.path.
Created symlink /etc/systemd/system/printer.target.wants/cups.service → /lib/systemd/system/cups.service.
Created symlink /etc/systemd/system/sockets.target.wants/cups.socket → /lib/systemd/system/cups.socket.
Setting up libio-html-perl (1.001-1) ...
Setting up libunity-gtk3-parser0:arm64 (0.0.0+18.04.20171202-0ubuntu2) ...
Setting up gnome-bluetooth (3.34.3-0ubuntu1) ...
Setting up libgnome-panel0 (1:3.36.2-0ubuntu1) ...

.
.
.

Setting up xarchiver (1:0.5.4.14-1) ...
Setting up lm-sensors (1:3.6.0-2ubuntu1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/lm-sensors.service → /lib/systemd/system/lm-
sensors.service.
Setting up libtimedate-perl (2.3200-1) ...
Setting up libnotify-bin (0.7.9-1ubuntu2) ...
Setting up libutempter0:arm64 (1.1.6-4) ...
Setting up liblightdm-gobject-1-0:arm64 (1.30.0-0ubuntu4~20.04.1) ...
Setting up libglibmm-2.4-1v5:arm64 (2.64.2-1) ...
Setting up cups-browsed (1.27.4-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/cups-browsed.service → /lib/systemd/system/cups-
browsed.service.
Setting up indicator-appmenu (15.02.0+16.10.20160927-0ubuntu5) ...
Setting up xfce4-cpufreq-plugin (1.2.1-1) ...
Setting up xfce4-fsguard-plugin:arm64 (1.1.1-1) ...
Setting up libhtml-form-perl (6.07-1) ...
Setting up lightdm (1.30.0-0ubuntu4~20.04.1) ...
Adding group `lightdm' (GID 126) ...
Done.
Adding system user `lightdm' (UID 118) ...
Adding new user `lightdm' (UID 118) with group `lightdm' ...
Creating home directory `/var/lib/lightdm' ...
usermod: no changes
usermod: no changes

```

```

Adding group `nopasswdlogin' (GID 127) ...
Done.
Setting up libxfce4ui-utils (4.14.1-1ubuntu1) ...
Setting up xfce4-smartbookmark-plugin (0.5.1-1) ...
Setting up libxfce4ui-1-0:arm64 (4.14.1-1ubuntu1) ...
    .
    .
    .
Setting up indicator-sound (12.10.2+18.10.20180612-0ubuntu1) ...
Setting up unity-settings-daemon (15.04.1+20.04.20200325-0ubuntu1) ...
Setting up indicator-session (17.3.20+19.10.20190921-0ubuntu1) ...
Setting up indicator-datetime (15.10+20.04.20200217-0ubuntu1) ...
Setting up unity-gtk3-module:arm64 (0.0.0+18.04.20171202-0ubuntu2) ...
Setting up xfce4-goodies (4.14.0) ...
Setting up unity-greeter (18.04.0+20.04.20200312-0ubuntu2) ...
Processing triggers for initramfs-tools (0.136ubuntu6.3) ...
update-initramfs: Generating /boot/initrd.img-3.16.85-65

```

6.1.1.6 Instalar VNC

```

Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  tightvnc-java
The following NEW packages will be installed:
  tightvncserver
0 upgraded, 1 newly installed, 0 to remove and 91 not upgraded.
Need to get 617 kB of archives.
After this operation, 1793 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 tightvncserver arm64 1.3.10-0ubuntu5 [617 kB]
Fetched 617 kB in 1s (617 kB/s)
Selecting previously unselected package tightvncserver.
(Reading database ... 98999 files and directories currently installed.)
Preparing to unpack ../tightvncserver_1.3.10-0ubuntu5_arm64.deb ...
Unpacking tightvncserver (1.3.10-0ubuntu5) ...
Setting up tightvncserver (1.3.10-0ubuntu5) ...
update-alternatives: using /usr/bin/tightvncserver to provide /usr/bin/vncserver (vncserver) in auto mode
update-alternatives: using /usr/bin/Xtightvnc to provide /usr/bin/Xvnc (Xvnc) in auto mode
update-alternatives: using /usr/bin/tightvncpasswd to provide /usr/bin/vncpasswd (vncpasswd) in auto mode
Processing triggers for man-db (2.9.1-1) ...

```

6.1.1.7 Instalar Midori

```

Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  gnome-icon-theme libpeas-1.0-0 libpeas-common
The following NEW packages will be installed:
  gnome-icon-theme libpeas-1.0-0 libpeas-common midori
0 upgraded, 4 newly installed, 0 to remove and 91 not upgraded.
Need to get 10.4 MB of archives.
After this operation, 20.0 MB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 gnome-icon-theme all 3.12.0-3 [9620 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 libpeas-common all 1.26.0-2 [9976 B]
Get:3 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 libpeas-1.0-0 arm64 1.26.0-2 [57.5 kB]
Get:4 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 midori arm64 7.0-2.1 [688 kB]
Fetched 10.4 MB in 2s (4380 kB/s)
Selecting previously unselected package gnome-icon-theme.
(Reading database ... 99018 files and directories currently installed.)
Preparing to unpack .../gnome-icon-theme_3.12.0-3_all.deb ...
Unpacking gnome-icon-theme (3.12.0-3) ...
Selecting previously unselected package libpeas-common.
Preparing to unpack .../libpeas-common_1.26.0-2_all.deb ...
Unpacking libpeas-common (1.26.0-2) ...
Selecting previously unselected package libpeas-1.0-0:arm64.
Preparing to unpack .../libpeas-1.0-0_1.26.0-2_arm64.deb ...
Unpacking libpeas-1.0-0:arm64 (1.26.0-2) ...
Selecting previously unselected package midori.
Preparing to unpack .../midori_7.0-2.1_arm64.deb ...
Unpacking midori (7.0-2.1) ...
Setting up libpeas-common (1.26.0-2) ...
Setting up gnome-icon-theme (3.12.0-3) ...
update-alternatives: using /usr/share/icons/gnome/scalable/places/debian-swirl.svg to provide
/usr/share/icons/gnome/scalable/places/start-here.svg (start-here.svg) in auto mode
Setting up libpeas-1.0-0:arm64 (1.26.0-2) ...
Setting up midori (7.0-2.1) ...
update-alternatives: using /usr/bin/midori to provide /usr/bin/x-www-browser (x-www-browser) in auto mode
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for desktop-file-utils (0.24-1ubuntu3) ...

```

6.1.1.8 Instalar NFS

```

Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  keyutils libevent-2.1-7 libnfsidmap2 libtirpc-common libtirpc3
  nfs-common rpcbind
Suggested packages:
  open-iscsi watchdog
The following NEW packages will be installed:
  keyutils libevent-2.1-7 libnfsidmap2 libtirpc-common libtirpc3
  nfs-common nfs-kernel-server rpcbind
0 upgraded, 8 newly installed, 0 to remove and 91 not upgraded.
Need to get 611 kB of archives.
After this operation, 2195 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 libtirpc-common all 1.2.5-1 [7632 B]
Get:2 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 libtirpc3 arm64 1.2.5-1 [74.1 kB]
Get:3 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 rpcbind arm64 1.2.5-8 [41.1 kB]
Get:4 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 keyutils arm64 1.6-6ubuntu1 [43.6 kB]
Get:5 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 libevent-2.1-7 arm64 2.1.11-stable-1 [125 kB]
Get:6 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 libnfsidmap2 arm64 0.25-5.1ubuntu1 [27.1 kB]
Get:7 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 nfs-common arm64 1:1.3.4-2.5ubuntu3.4
[197 kB]
Get:8 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 nfs-kernel-server arm64 1:1.3.4-
2.5ubuntu3.4 [94.9 kB]
Fetched 611 kB in 1s (640 kB/s)
Selecting previously unselected package libtirpc-common.
(Reading database ... 105189 files and directories currently installed.)
Preparing to unpack .../0-libtirpc-common_1.2.5-1_all.deb ...
Unpacking libtirpc-common (1.2.5-1) ...
      .
      .
      .
Selecting previously unselected package nfs-kernel-server.
Preparing to unpack .../7-nfs-kernel-server_1%3a1.3.4-2.5ubuntu3.4_arm64.deb ...
Unpacking nfs-kernel-server (1:1.3.4-2.5ubuntu3.4) ...
Setting up libtirpc-common (1.2.5-1) ...
Setting up libevent-2.1-7:arm64 (2.1.11-stable-1) ...
Setting up keyutils (1.6-6ubuntu1) ...
Setting up libnfsidmap2:arm64 (0.25-5.1ubuntu1) ...
Setting up libtirpc3:arm64 (1.2.5-1) ...
Setting up rpcbind (1.2.5-8) ...
Created symlink /etc/systemd/system/multi-user.target.wants/rpcbind.service →
/lib/systemd/system/rpcbind.service.

```

```

Created symlink /etc/systemd/system/sockets.target.wants/rpcbind.socket → /lib/systemd/system/rpcbind.socket.
Setting up nfs-common (1:1.3.4-2.5ubuntu3.4) ...
Creating config file /etc/idmapd.conf with new version
Adding system user `statd' (UID 120) ...
Adding new user `statd' (UID 120) with group `nogroup' ...
Not creating home directory `/var/lib/nfs'.
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-client.target → /lib/systemd/system/nfs-client.target.
Created symlink /etc/systemd/system/remote-fs.target.wants/nfs-client.target → /lib/systemd/system/nfs-client.target.
nfs-utils.service is a disabled or a static unit, not starting it.
Setting up nfs-kernel-server (1:1.3.4-2.5ubuntu3.4) ...
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-server.service → /lib/systemd/system/nfs-server.service.
Job for nfs-server.service canceled.
Creating config file /etc/exports with new version
Creating config file /etc/default/nfs-kernel-server with new version
Processing triggers for systemd (245.4-4ubuntu3.4) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.1) ...
exporting 172.16.0.0/24:/home
Synchronizing state of nfs-kernel-server.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable nfs-kernel-server

```

6.1.1.9 Instalar Munge

```

Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libmunge2
The following NEW packages will be installed:
  libmunge2 munge
0 upgraded, 2 newly installed, 0 to remove and 91 not upgraded.
Need to get 105 kB of archives.
After this operation, 385 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 libmunge2 arm64 0.5.13-2build1 [19.4 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 munge arm64 0.5.13-2build1 [86.0 kB]
Fetched 105 kB in 1s (148 kB/s)
Selecting previously unselected package libmunge2.
(Reading database ... 105354 files and directories currently installed.)
Preparing to unpack .../libmunge2_0.5.13-2build1_arm64.deb ...
Unpacking libmunge2 (0.5.13-2build1) ...
Selecting previously unselected package munge.
Preparing to unpack .../munge_0.5.13-2build1_arm64.deb ...

```

```

Unpacking munge (0.5.13-2build1) ...
Setting up libmunge2 (0.5.13-2build1) ...
Setting up munge (0.5.13-2build1) ...
Generating a pseudo-random key using /dev/urandom completed.
Please refer to /usr/share/doc/munge/README.Debian for instructions to generate more secure key.
Created          symlink          /etc/systemd/system/multi-user.target.wants/munge.service      →
/lib/systemd/system/munge.service.
Processing triggers for systemd (245.4-4ubuntu3.4) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.1) ...
Synchronizing state of munge.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable munge
Generating a pseudo-random key using /dev/urandom completed.
2+0 records in
2+0 records out
1024 bytes (1.0 kB, 1.0 KiB) copied, 0.000294005 s, 3.5 MB/s

```

6.1.1.10 Instalar Slurm

Reading package lists... Done

Building dependency tree

Reading state information... Done

The following additional packages will be installed:

```

freeipmi-common gcc-10-base libaec0 libatomic1 libcc1-0 libcurl4
libdbi1 libfreeipmi17 libgcc-s1 libgcc1 libgfortran5 libgomp1
libhdf5-103 libhwloc-plugins libhwloc15 libipmimonitoring6 libitm1
liblsan0 liblua5.1-0 libnuma1 librrd8 libstdc++6 libsz2 libtsan0
libubsan1 ocl-icd-libopencl1 slurm-client slurm-wlm-basic-plugins
slurmctld slurmd

```

Suggested packages:

```

freeipmi-tools opencl-icd

```

The following NEW packages will be installed:

```

freeipmi-common libaec0 libcurl4 libdbi1 libfreeipmi17 libgfortran5
libhdf5-103 libhwloc-plugins libhwloc15 libipmimonitoring6 liblua5.1-0
libnuma1 librrd8 libsz2 ocl-icd-libopencl1 slurm-client slurm-wlm
slurm-wlm-basic-plugins slurmctld slurmd

```

The following packages will be upgraded:

```

gcc-10-base libatomic1 libcc1-0 libgcc-s1 libgcc1 libgomp1 libitm1
liblsan0 libstdc++6 libtsan0 libubsan1

```

11 upgraded, 20 newly installed, 0 to remove and 80 not upgraded.

Need to get 10.1 MB of archives.

After this operation, 36.8 MB of additional disk space will be used.

```

Get:1 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 freeipmi-common all 1.6.4-3ubuntu1.1
[179 kB]

```

Get:2 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 libatomic1 arm64 10.3.0-1ubuntu1~20.04 [9820 B]

•
•
•

Get:30 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 ocl-icd-libopencl1 arm64 2.2.11-1ubuntu1 [29.2 kB]

Get:31 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 libhwloc-plugins arm64 2.1.0+dfsg-4 [13.8 kB]

Fetches 10.1 MB in 3s (3731 kB/s)

Extracting templates from packages: 100%

Selecting previously unselected package freeipmi-common.

(Reading database ... 105392 files and directories currently installed.)

Preparing to unpack .../freeipmi-common_1.6.4-3ubuntu1.1_all.deb ...

Unpacking freeipmi-common (1.6.4-3ubuntu1.1) ...

Preparing to unpack .../libatomic1_10.3.0-1ubuntu1~20.04_arm64.deb ...

Unpacking libatomic1:arm64 (10.3.0-1ubuntu1~20.04) over (10.2.0-5ubuntu1~20.04) ...

Preparing to unpack .../libubsan1_10.3.0-1ubuntu1~20.04_arm64.deb ...

Unpacking libubsan1:arm64 (10.3.0-1ubuntu1~20.04) over (10.2.0-5ubuntu1~20.04) ...

•
•
•

Selecting previously unselected package libhwloc-plugins:arm64.

Preparing to unpack .../18-libhwloc-plugins_2.1.0+dfsg-4_arm64.deb ...

Unpacking libhwloc-plugins:arm64 (2.1.0+dfsg-4) ...

Setting up libdbi1:arm64 (0.9.0-5) ...

Setting up libgcc1 (1:10.3.0-1ubuntu1~20.04) ...

Setting up freeipmi-common (1.6.4-3ubuntu1.1) ...

•
•
•

Setting up slurm-wlm-basic-plugins (19.05.5-1) ...

Setting up slurmd (19.05.5-1) ...

update-alternatives: using /usr/sbin/slurmd-wlm to provide /usr/sbin/slurmd (slurmd) in auto mode

Created symlink /etc/systemd/system/multi-user.target.wants/slurmd.service → /lib/systemd/system/slurmd.service.

Setting up slurm-client (19.05.5-1) ...

Setting up slurmctld (19.05.5-1) ...

update-alternatives: using /usr/sbin/slurmctld-wlm to provide /usr/sbin/slurmctld (slurmctld) in auto mode

Created symlink /etc/systemd/system/multi-user.target.wants/slurmctld.service → /lib/systemd/system/slurmctld.service.

Setting up slurm-wlm (19.05.5-1) ...

Processing triggers for systemd (245.4-4ubuntu3.4) ...

Processing triggers for man-db (2.9.1-1) ...

Processing triggers for libc-bin (2.31-0ubuntu9.1) ...

Synchronizing state of slurmctld.service with SysV service script with /lib/systemd/systemd-sysv-install.

Executing: /lib/systemd/systemd-sysv-install enable slurmctld

A continuación se muestra el estado del servicio de slurm para constatar que se inicia correctamente:

```
root@odroid:~/odroid-cluster# systemctl status slurmctld
```

- slurmctld.service - Slurm controller daemon

Loaded: loaded (/lib/systemd/system/slurmctld.service; enabled; vendor preset: enabled)

Active: active (running) since Wed 2021-07-21 20:06:20 CEST; 30min ago

Docs: man:slurmctld(8)

Main PID: 35624 (slurmctld)

Memory: 4.1M

CGroup: /system.slice/slurmctld.service

└─35624 /usr/sbin/slurmctld

```
Jul 21 20:06:20 master systemd[1]: Starting Slurm controller daemon...
```

```
Jul 21 20:06:20 master systemd[1]: Started Slurm controller daemon.
```

6.1.1.11 Instalar Mpich

Reading package lists... Done

Building dependency tree

Reading state information... Done

The following additional packages will be installed:

gfortran gfortran-9 hwloc-nox libgfortran-9-dev libmpich-dev
libmpich12

Suggested packages:

gfortran-doc gfortran-9-doc libcoarrays-dev mpich-doc

The following NEW packages will be installed:

gfortran gfortran-9 hwloc-nox libgfortran-9-dev libmpich-dev
libmpich12 mpich

0 upgraded, 7 newly installed, 0 to remove and 80 not upgraded.

Need to get 9912 kB of archives.

After this operation, 42.0 MB of additional disk space will be used.

```
Get:1 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 libgfortran-9-dev arm64 9.3.0-17ubuntu1~20.04 [390 kB]
```

```
Get:2 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 gfortran-9 arm64 9.3.0-17ubuntu1~20.04 [6373 kB]
```

```
Get:3 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 gfortran arm64 4:9.3.0-1ubuntu2 [1360 B]
```

```
Get:4 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 hwloc-nox arm64 2.1.0+dfsg-4 [145 kB]
```

```
Get:5 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 libmpich12 arm64 3.3.2-2build1 [1005 kB]
```

```
Get:6 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 mpich arm64 3.3.2-2build1 [345 kB]
```

```
Get:7 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 libmpich-dev arm64 3.3.2-2build1 [1654 kB]
```

Fetches 9912 kB in 2s (4187 kB/s)

```

Selecting previously unselected package libgfortran-9-dev:arm64.
(Reading database ... 105752 files and directories currently installed.)
Preparing to unpack .../0-libgfortran-9-dev_9.3.0-17ubuntu1~20.04_arm64.deb ...
Unpacking libgfortran-9-dev:arm64 (9.3.0-17ubuntu1~20.04) ...
Selecting previously unselected package gfortran-9.
Preparing to unpack .../1-gfortran-9_9.3.0-17ubuntu1~20.04_arm64.deb ...
Unpacking gfortran-9 (9.3.0-17ubuntu1~20.04) ...
Selecting previously unselected package gfortran.
Preparing to unpack .../2-gfortran_4%3a9.3.0-1ubuntu2_arm64.deb ...
Unpacking gfortran (4:9.3.0-1ubuntu2) ...
Selecting previously unselected package hwloc-nox.
Preparing to unpack .../3-hwloc-nox_2.1.0+dfsg-4_arm64.deb ...
Unpacking hwloc-nox (2.1.0+dfsg-4) ...
Selecting previously unselected package libmpich12:arm64.
Preparing to unpack .../4-libmpich12_3.3.2-2build1_arm64.deb ...
Unpacking libmpich12:arm64 (3.3.2-2build1) ...
Selecting previously unselected package mpich.
Preparing to unpack .../5-mpich_3.3.2-2build1_arm64.deb ...
Unpacking mpich (3.3.2-2build1) ...
Selecting previously unselected package libmpich-dev:arm64.
Preparing to unpack .../6-libmpich-dev_3.3.2-2build1_arm64.deb ...
Unpacking libmpich-dev:arm64 (3.3.2-2build1) ...
Setting up hwloc-nox (2.1.0+dfsg-4) ...
Setting up libgfortran-9-dev:arm64 (9.3.0-17ubuntu1~20.04) ...
Setting up libmpich12:arm64 (3.3.2-2build1) ...
Setting up gfortran-9 (9.3.0-17ubuntu1~20.04) ...
Setting up mpich (3.3.2-2build1) ...
update-alternatives: using /usr/bin/mpicc.mpich to provide /usr/bin/mpicc (mpi) in auto mode
update-alternatives: using /usr/bin/mpirun.mpich to provide /usr/bin/mpirun (mpirun) in auto mode
Setting up gfortran (4:9.3.0-1ubuntu2) ...
update-alternatives: using /usr/bin/gfortran to provide /usr/bin/f95 (f95) in auto mode
update-alternatives: using /usr/bin/gfortran to provide /usr/bin/f77 (f77) in auto mode
Setting up libmpich-dev:arm64 (3.3.2-2build1) ...
update-alternatives: using /usr/include/aarch64-linux-gnu/mpich to provide /usr/include/aarch64-linux-gnu/mpi
(mpi-aarch64-linux-gnu) in auto mode
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.1) ...

```

6.1.1.12 Instalar dnsmasq

```

RTNETLINK answers: File exists
Reading package lists... Done
Building dependency tree
Reading state information... Done

```

Suggested packages:
 resolvconf

The following NEW packages will be installed:
 dnsmasq

0 upgraded, 1 newly installed, 0 to remove and 80 not upgraded.
 Need to get 16.5 kB of archives.
 After this operation, 75.8 kB of additional disk space will be used.
 Get:1 http://ports.ubuntu.com/ubuntu-ports focal-updates/universe arm64 dnsmasq all 2.80-1.1ubuntu1.4 [16.5 kB]

Fetches 16.5 kB in 0s (42.7 kB/s)
 Selecting previously unselected package dnsmasq.
 (Reading database ... 105913 files and directories currently installed.)
 Preparing to unpack .../dnsmasq_2.80-1.1ubuntu1.4_all.deb ...
 Unpacking dnsmasq (2.80-1.1ubuntu1.4) ...
 Setting up dnsmasq (2.80-1.1ubuntu1.4) ...

```
Created symlink /etc/systemd/system/multi-user.target.wants/dnsmasq.service →
/lib/systemd/system/dnsmasq.service.
```

Job for dnsmasq.service failed because the control process exited with error code.
 See "systemctl status dnsmasq.service" and "journalctl -xe" for details.
 invoke-rc.d: initscript dnsmasq, action "start" failed.

- dnsmasq.service - dnsmasq - A lightweight DHCP and caching DNS server
 - Loaded: loaded (/lib/systemd/system/dnsmasq.service; enabled; vendor preset: enabled)
 - Active: failed (Result: exit-code) since Wed 2021-07-21 20:05:54 CEST; 26ms ago
 - Process: 34818 ExecStartPre=/usr/sbin/dnsmasq --test (code=exited, status=0/SUCCESS)
 - Process: 34820 ExecStart=/etc/init.d/dnsmasq systemd-exec (code=exited, status=2)

Jul 21 20:05:54 master systemd[1]: Starting dnsmasq - A lightweight DHCP and caching DNS server...
 Jul 21 20:05:54 master dnsmasq[34818]: dnsmasq: syntax check OK.
 Jul 21 20:05:54 master dnsmasq[34820]: dnsmasq: failed to create listening socket for port 53: Address already in use
 Jul 21 20:05:54 master dnsmasq[34820]: failed to create listening socket for port 53: Address already in use
 Jul 21 20:05:54 master dnsmasq[34820]: FAILED to start up
 Jul 21 20:05:54 master systemd[1]: dnsmasq.service: Control process exited, code=exited, status=2/INVALIDARGUMENT
 Jul 21 20:05:54 master systemd[1]: dnsmasq.service: Failed with result 'exit-code'.
 Jul 21 20:05:54 master systemd[1]: Failed to start dnsmasq - A lightweight DHCP and caching DNS server.
 Processing triggers for systemd (245.4-4ubuntu3.4) ...
 Synchronizing state of dnsmasq.service with SysV service script with /lib/systemd/systemd-sysv-install.
 Executing: /lib/systemd/systemd-sysv-install disable dnsmasq
 Removed /etc/systemd/system/multi-user.target.wants/dnsmasq.service.
 Removed /etc/systemd/system/multi-user.target.wants/systemd-resolved.service.
 Removed /etc/systemd/system/dbus-org.freedesktop.resolve1.service.
 run-parts: failed to stat component /etc/network/if-post-down.d/avahi-daemon: No such file or directory
 cp: missing destination file operand after '/etc/dnsmasq.conf'
 Try 'cp --help' for more information.

A continuación se muestra el estado del servicio de dnsmasq para corroborar que se inicia correctamente y que ejecuta sus funciones como servidor DHCP:

```
root@odroid:~/odroid-cluster# systemctl status dnsmasq
● dnsmasq.service - dnsmasq - A lightweight DHCP and caching DNS server
   Loaded: loaded (/lib/systemd/system/dnsmasq.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2021-07-21 20:05:59 CEST; 20min ago
 Main PID: 35185 (dnsmasq)
    Memory: 736.0K
    CGroup: /system.slice/dnsmasq.service
           └─35185 /usr/sbin/dnsmasq -x /run/dnsmasq/dnsmasq.pid -u dnsmasq -7 /etc/dnsmasq.d,.dpkg-dist,.dpkg-old,.dpkg-
new --local-service --tru>
           └─35186 /usr/sbin/dnsmasq -x /run/dnsmasq/dnsmasq.pid -u dnsmasq -7 /etc/dnsmasq.d,.dpkg-dist,.dpkg-old,.dpkg-
new --local-service --tru>

Jul 21 20:06:20 master dnsmasq-dhcp[35185]: DHCPDISCOVER(eth0) 00:1e:06:33:4d:fa
Jul 21 20:06:20 master dnsmasq-dhcp[35185]: DHCPOFFER(eth0) 172.16.0.231 00:1e:06:33:4d:fa
Jul 21 20:06:20 master dnsmasq-dhcp[35185]: DHCPREQUEST(eth0) 172.16.0.41 00:1e:06:33:ce:35
Jul 21 20:06:20 master dnsmasq-dhcp[35185]: DHCPACK(eth0) 172.16.0.41 00:1e:06:33:ce:35 odroid
Jul 21 20:06:20 master dnsmasq-dhcp[35185]: DHCPREQUEST(eth0) 172.16.0.231 00:1e:06:33:4d:fa
Jul 21 20:06:20 master dnsmasq-dhcp[35185]: DHCPACK(eth0) 172.16.0.231 00:1e:06:33:4d:fa odroid
Jul 21 20:06:20 master dnsmasq[35185]: inotify, new or changed file /etc/hosts.d/tmp_hosts
Jul 21 20:06:20 master dnsmasq[35185]: read /etc/hosts.d/tmp_hosts - 1 addresses
Jul 21 20:06:20 master dnsmasq[35185]: inotify, new or changed file /etc/hosts.d/tmp_hosts
Jul 21 20:06:20 master dnsmasq[35185]: read /etc/hosts.d/tmp_hosts - 2 addresses
```

6.1.1.13 Instalar Ganglia

Reading package lists... Done

Building dependency tree

Reading state information... Done

unzip is already the newest version (6.0-25ubuntu1).

unzip set to manually installed.

wget is already the newest version (1.20.3-1ubuntu1).

wget set to manually installed.

The following additional packages will be installed:

apache2-bin apache2-data apache2-utils galera-3 gawk libaio1

libapache2-mod-php7.4 libapr1 libaprutil1 libaprutil1-dbd-sqlite3

libaprutil1-ldap libcgi-fast-perl libcgi-pm-perl

libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl libfcgi-perl

libhtml-template-perl liblua5.2-0 libonig5 libreadline5 libsigsegv2

libsnapy1v5 libterm-readkey-perl libzip5 mariadb-client-10.3

mariadb-client-core-10.3 mariadb-common mariadb-server-10.3

mariadb-server-core-10.3 php-common php7.4 php7.4-cli php7.4-common

php7.4-curl php7.4-gd php7.4-json php7.4-mbstring php7.4-mysql

php7.4-opcache php7.4-readline php7.4-zip rsync socat

Suggested packages:

apache2-doc apache2-suexec-pristine | apache2-suexec-custom ufw
gawk-doc php-pear libclone-perl libmldbm-perl libnet-daemon-perl
libsql-statement-perl libipc-sharedcache-perl mailx mariadb-test
tinycd

The following NEW packages will be installed:

apache2 apache2-bin apache2-data apache2-utils galera-3 gawk libaio1
libapache2-mod-php libapache2-mod-php7.4 libapr1 libaprutil1
libaprutil1-dbd-sqlite3 libaprutil1-ldap libcgi-fast-perl
libcgi-pm-perl libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl
libfcgi-perl libhtml-template-perl liblua5.2-0 libonig5 libreadline5
libsigsegv2 libsnappy1v5 libterm-readkey-perl libzip5
mariadb-client-10.3 mariadb-client-core-10.3 mariadb-common
mariadb-server mariadb-server-10.3 mariadb-server-core-10.3 php
php-common php-curl php-gd php-mbstring php-mysql php-zip php7.4
php7.4-cli php7.4-common php7.4-curl php7.4-gd php7.4-json
php7.4-mbstring php7.4-mysql php7.4-opcache php7.4-readline php7.4-zip
rsync socat

0 upgraded, 53 newly installed, 0 to remove and 80 not upgraded.

Need to get 25.3 MB of archives.

After this operation, 192 MB of additional disk space will be used.

Get:1 <http://ports.ubuntu.com/ubuntu-ports focal/main arm64 rsync arm64 3.1.3-8> [308 kB]

Get:2 <http://ports.ubuntu.com/ubuntu-ports focal/main arm64 libapr1 arm64 1.6.5-1ubuntu1> [85.4 kB]

Get:3 <http://ports.ubuntu.com/ubuntu-ports focal/main arm64 libaprutil1 arm64 1.6.1-4ubuntu2> [82.4 kB]

.
.
.

Get:51 <http://ports.ubuntu.com/ubuntu-ports focal/main arm64 php-mysql all 2:7.4+75> [2000 B]

Get:52 <http://ports.ubuntu.com/ubuntu-ports focal-updates/universe arm64 php7.4-zip arm64 7.4.3-4ubuntu2.5>
[20.1 kB]

Get:53 <http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 php-zip all 2:7.4+75> [1996 B]

Fetches 25.3 MB in 5s (5479 kB/s)

Extracting templates from packages: 100%

Preconfiguring packages ...

Selecting previously unselected package rsync.

(Reading database ... 105930 files and directories currently installed.)

Preparing to unpack .../00-rsync_3.1.3-8_arm64.deb ...

Unpacking rsync (3.1.3-8) ...

.
.
.

Selecting previously unselected package php-zip.

```

Preparing to unpack .../29-php-zip_2%3a7.4+75_all.deb ...
Unpacking php-zip (2:7.4+75) ...
Setting up libconfig-inifiles-perl (3.000002-1) ...
Setting up php-common (2:75) ...
Created symlink /etc/systemd/system/timers.target.wants/phpsessionclean.timer
/lib/systemd/system/phpsessionclean.timer. →
Setting up libreadline5:arm64 (5.2+dfsg-3build3) ...
Setting up php7.4-common (7.4.3-4ubuntu2.5) ...
Creating config file /etc/php/7.4/mods-available/calendar.ini with new version
Creating config file /etc/php/7.4/mods-available/ctype.ini with new version
.
.
.
Creating config file /etc/php/7.4/mods-available/sysvsem.ini with new version
Creating config file /etc/php/7.4/mods-available/sysvshm.ini with new version
Creating config file /etc/php/7.4/mods-available/tokenizer.ini with new version
Setting up php7.4-curl (7.4.3-4ubuntu2.5) ...
Creating config file /etc/php/7.4/mods-available/curl.ini with new version
Setting up php7.4-mysql (7.4.3-4ubuntu2.5) ...
Creating config file /etc/php/7.4/mods-available/mysqlnd.ini with new version
Creating config file /etc/php/7.4/mods-available/mysqli.ini with new version
Creating config file /etc/php/7.4/mods-available/pdo_mysql.ini with new version
Setting up libzip5:arm64 (1.5.1-0ubuntu1) ...
.
.
.
Setting up mariadb-client-10.3 (1:10.3.29-0ubuntu0.20.04.1) ...
Setting up mariadb-server-10.3 (1:10.3.29-0ubuntu0.20.04.1) ...
Created symlink /etc/systemd/system/mysql.service → /lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/mysqld.service → /lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service
/lib/systemd/system/mariadb.service. →
Setting up libapache2-mod-php7.4 (7.4.3-4ubuntu2.5) ...
Package apache2 is not configured yet. Will defer actions by package libapache2-mod-php7.4.
Creating config file /etc/php/7.4/apache2/php.ini with new version
No module matches
Setting up apache2 (2.4.41-4ubuntu3.4) ...
Enabling module mpm_event.
Enabling module authz_core.
.
.
.
Enabling site 000-default.

```

```

info: Switch to mpm prefork for package libapache2-mod-php7.4
Module mpm_event disabled.
Enabling module mpm_prefork.
info: Executing deferred 'a2enmod php7.4' for package libapache2-mod-php7.4
Enabling module php7.4.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service →
/lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service →
/lib/systemd/system/apache-htcacheclean.service.
Setting up mariadb-server (1:10.3.29-0ubuntu0.20.04.1) ...
Setting up php7.4 (7.4.3-4ubuntu2.5) ...
Setting up libapache2-mod-php (2:7.4+75) ...
Setting up php (2:7.4+75) ...
Processing triggers for systemd (245.4-4ubuntu3.4) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.1) ...
Processing triggers for php7.4-cli (7.4.3-4ubuntu2.5) ...
Processing triggers for libapache2-mod-php7.4 (7.4.3-4ubuntu2.5) ...
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libconfuse-common libconfuse2 libganglia1 php-xml php7.4-xml
Suggested packages:
  librrds-perl
The following NEW packages will be installed:
  ganglia-monitor ganglia-webfrontend gmetad libconfuse-common
  libconfuse2 libganglia1 php-xml php7.4-xml rrdtool
0 upgraded, 9 newly installed, 0 to remove and 80 not upgraded.
Need to get 2281 kB of archives.
After this operation, 15.0 MB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 libconfuse-common all 3.2.2+dfsg-1 [5512 B]
Get:2 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 libconfuse2 arm64 3.2.2+dfsg-1 [22.6 kB]
.
.
.
Preparing to unpack .../8-gmetad_3.6.0-7ubuntu4_arm64.deb ...
Unpacking gmetad (3.6.0-7ubuntu4) ...
Setting up php7.4-xml (7.4.3-4ubuntu2.5) ...
Creating config file /etc/php/7.4/mods-available/dom.ini with new version
Creating config file /etc/php/7.4/mods-available/simplexml.ini with new version
Creating config file /etc/php/7.4/mods-available/xml.ini with new version
Creating config file /etc/php/7.4/mods-available/xmlreader.ini with new version

```

```

Creating config file /etc/php/7.4/mods-available/xmlwriter.ini with new version
Creating config file /etc/php/7.4/mods-available/xsl.ini with new version
Setting up libconfuse-common (3.2.2+dfsg-1) ...
Setting up libconfuse2:arm64 (3.2.2+dfsg-1) ...
Setting up rrdtool (1.7.2-3build2) ...
Setting up php-xml (2:7.4+75) ...
Setting up ganglia-webfrontend (3.6.1-3) ...
Setting up libganglia1 (3.6.0-7ubuntu4) ...
Setting up gmetad (3.6.0-7ubuntu4) ...
Adding system user: ganglia.
Setting up ganglia-monitor (3.6.0-7ubuntu4) ...
Processing triggers for systemd (245.4-4ubuntu3.4) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libapache2-mod-php7.4 (7.4.3-4ubuntu2.5) ...
Processing triggers for libc-bin (2.31-0ubuntu9.1) ...
Processing triggers for php7.4-cli (7.4.3-4ubuntu2.5) ...
gmetad.service is not a native service, redirecting to systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable gmetad
ganglia-monitor.service is not a native service, redirecting to systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable ganglia-monitor
Ganglia correctly configured, go to http://your_server_ip/ganglia to access ganglia dashboard

```

6.1.1.14 Función add_scripts

En la función `add_scripts` únicamente se puede observar por el terminal el momento en que se descomprime el archivo `bin.tgz` que contiene los scripts de administración remota.

```

bin/
bin/global_execute_seq
bin/global_execute_par

```

6.1.1.15 Configurar idioma

```

Generating locales (this might take a while)...
  es_ES.UTF-8... done
Generation complete.
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  language-pack-es-base language-pack-gnome-es-base
Se instalarán los siguientes paquetes NUEVOS:
  gnome-user-docs-es language-pack-es language-pack-es-base
  language-pack-gnome-es language-pack-gnome-es-base wspanish
0 actualizados, 6 nuevos se instalarán, 0 para eliminar y 80 no actualizados.
Se necesita descargar 6.586 kB de archivos.

```

Se utilizarán 29,1 MB de espacio de disco adicional después de esta operación.

Des:1 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 language-pack-es-base all 1:20.04+20210121 [2.924 kB]

Des:2 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 language-pack-es all 1:20.04+20210121 [1.900 B]

Des:3 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 language-pack-gnome-es-base all 1:20.04+20210121 [2.992 kB]

Des:4 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 language-pack-gnome-es all 1:20.04+20210121 [1.924 B]

Des:5 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 gnome-user-docs-es all 3.36.2+git20200704-0ubuntu0.1 [455 kB]

Des:6 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 wspanish all 1.0.28 [211 kB]

Descargados 6.586 kB en 3s (2.073 kB/s)

Preconfigurando paquetes ...

Seleccionando el paquete language-pack-es-base previamente no seleccionado.

(Leyendo la base de datos ... 109075 ficheros o directorios instalados actualmente.)

Preparando para desempaquetar .../0-language-pack-es-base_1%3a20.04+20210121_all.deb ...

Desempaquetando language-pack-es-base (1:20.04+20210121) ...

Seleccionando el paquete language-pack-es previamente no seleccionado.

Preparando para desempaquetar .../1-language-pack-es_1%3a20.04+20210121_all.deb ...

Desempaquetando language-pack-es (1:20.04+20210121) ...

Seleccionando el paquete language-pack-gnome-es-base previamente no seleccionado.

Preparando para desempaquetar .../2-language-pack-gnome-es-base_1%3a20.04+20210121_all.deb ...

Desempaquetando language-pack-gnome-es-base (1:20.04+20210121) ...

Seleccionando el paquete language-pack-gnome-es previamente no seleccionado.

Preparando para desempaquetar .../3-language-pack-gnome-es_1%3a20.04+20210121_all.deb ...

Desempaquetando language-pack-gnome-es (1:20.04+20210121) ...

Seleccionando el paquete gnome-user-docs-es previamente no seleccionado.

Preparando para desempaquetar .../4-gnome-user-docs-es_3.36.2+git20200704-0ubuntu0.1_all.deb ...

Desempaquetando gnome-user-docs-es (3.36.2+git20200704-0ubuntu0.1) ...

Seleccionando el paquete wspanish previamente no seleccionado.

Preparando para desempaquetar .../5-wspanish_1.0.28_all.deb ...

Desempaquetando wspanish (1.0.28) ...

Configurando gnome-user-docs-es (3.36.2+git20200704-0ubuntu0.1) ...

Configurando wspanish (1.0.28) ...

Configurando language-pack-es (1:20.04+20210121) ...

Configurando language-pack-gnome-es (1:20.04+20210121) ...

Configurando language-pack-gnome-es-base (1:20.04+20210121) ...

Configurando language-pack-es-base (1:20.04+20210121) ...

Generating locales (this might take a while)...

es_AR.UTF-8... done

es_BO.UTF-8... done

es_CL.UTF-8... done

es_CO.UTF-8... done

es_CR.UTF-8... done

```

es_CU.UTF-8... done
es_DO.UTF-8... done
es_EC.UTF-8... done
es_GT.UTF-8... done
es_HN.UTF-8... done
es_MX.UTF-8... done
es_NI.UTF-8... done
es_PA.UTF-8... done
es_PE.UTF-8... done
es_PR.UTF-8... done
es_PY.UTF-8... done
es_SV.UTF-8... done
es_US.UTF-8... done
es_UY.UTF-8... done
es_VE.UTF-8... done

```

Generation complete.

Procesando disparadores para man-db (2.9.1-1) ...

Procesando disparadores para cracklib-runtime (2.9.6-3.2) ...

Procesando disparadores para dictionaries-common (1.28.1) ...

6.1.1.16 add_monitoring.sh

Leyendo lista de paquetes... Hecho

Creando árbol de dependencias

Leyendo la información de estado... Hecho

Se pueden actualizar 80 paquetes. Ejecute «apt list --upgradable» para verlos.

Leyendo lista de paquetes... Hecho

Creando árbol de dependencias

Leyendo la información de estado... Hecho

Se instalarán los siguientes paquetes adicionales:

```

libexpat1-dev libpython3-dev libpython3.8-dev python-pip-whl
python3-dev python3-distutils python3-lib2to3 python3-setuptools
python3-wheel python3.8-dev zlib1g-dev

```

Paquetes sugeridos:

```
python-setuptools-doc
```

Se instalarán los siguientes paquetes NUEVOS:

```

libexpat1-dev libpython3-dev libpython3.8-dev python-pip-whl
python3-dev python3-distutils python3-lib2to3 python3-pip
python3-setuptools python3-wheel python3.8-dev zlib1g-dev

```

0 actualizados, 12 nuevos se instalarán, 0 para eliminar y 80 no actualizados.

Se necesita descargar 7.139 kB de archivos.

Se utilizarán 27,1 MB de espacio de disco adicional después de esta operación.

Des:1 <http://ports.ubuntu.com/ubuntu-ports focal/main arm64 libexpat1-dev arm64 2.2.9-1build1> [103 kB]

Des:2 <http://ports.ubuntu.com/ubuntu-ports> focal-updates/main arm64 libpython3.8-dev arm64 3.8.10-0ubuntu1~20.04 [3.755 kB]

Des:3 <http://ports.ubuntu.com/ubuntu-ports> focal/main arm64 libpython3-dev arm64 3.8.2-0ubuntu2 [7.236 B]

Des:4 <http://ports.ubuntu.com/ubuntu-ports> focal-updates/universe arm64 python-pip-whl all 20.0.2-5ubuntu1.6 [1.805 kB]

Des:5 <http://ports.ubuntu.com/ubuntu-ports> focal-updates/main arm64 zlib1g-dev arm64 1:1.2.11.dfsg-2ubuntu1.2 [154 kB]

Des:6 <http://ports.ubuntu.com/ubuntu-ports> focal-updates/main arm64 python3.8-dev arm64 3.8.10-0ubuntu1~20.04 [511 kB]

Des:7 <http://ports.ubuntu.com/ubuntu-ports> focal-updates/main arm64 python3-lib2to3 all 3.8.10-0ubuntu1~20.04 [76,3 kB]

Des:8 <http://ports.ubuntu.com/ubuntu-ports> focal-updates/main arm64 python3-distutils all 3.8.10-0ubuntu1~20.04 [141 kB]

Des:9 <http://ports.ubuntu.com/ubuntu-ports> focal/main arm64 python3-dev arm64 3.8.2-0ubuntu2 [1.212 B]

Des:10 <http://ports.ubuntu.com/ubuntu-ports> focal/main arm64 python3-setuptools all 45.2.0-1 [330 kB]

Des:11 <http://ports.ubuntu.com/ubuntu-ports> focal/universe arm64 python3-wheel all 0.34.2-1 [23,8 kB]

Des:12 <http://ports.ubuntu.com/ubuntu-ports> focal-updates/universe arm64 python3-pip all 20.0.2-5ubuntu1.6 [231 kB]

Descargados 7.139 kB en 2s (3.545 kB/s)

Seleccionando el paquete libexpat1-dev:arm64 previamente no seleccionado.

(Leyendo la base de datos ... 111270 ficheros o directorios instalados actualmente.)

Preparando para desempaquetar .../00-libexpat1-dev_2.2.9-1build1_arm64.deb ...

Desempaquetando libexpat1-dev:arm64 (2.2.9-1build1) ...

Seleccionando el paquete libpython3.8-dev:arm64 previamente no seleccionado.

Preparando para desempaquetar .../01-libpython3.8-dev_3.8.10-0ubuntu1~20.04_arm64.deb ...

Desempaquetando libpython3.8-dev:arm64 (3.8.10-0ubuntu1~20.04) ...

Seleccionando el paquete libpython3-dev:arm64 previamente no seleccionado.

Preparando para desempaquetar .../02-libpython3-dev_3.8.2-0ubuntu2_arm64.deb ...

Desempaquetando libpython3-dev:arm64 (3.8.2-0ubuntu2) ...

Seleccionando el paquete python-pip-whl previamente no seleccionado.

Preparando para desempaquetar .../03-python-pip-whl_20.0.2-5ubuntu1.6_all.deb ...

Desempaquetando python-pip-whl (20.0.2-5ubuntu1.6) ...

Seleccionando el paquete zlib1g-dev:arm64 previamente no seleccionado.

Preparando para desempaquetar .../04-zlib1g-dev_1%3a1.2.11.dfsg-2ubuntu1.2_arm64.deb ...

Desempaquetando zlib1g-dev:arm64 (1:1.2.11.dfsg-2ubuntu1.2) ...

·
·
·

Leyendo lista de paquetes... Hecho

Creando árbol de dependencias

Leyendo la información de estado... Hecho

Paquetes sugeridos:

isag

Se instalarán los siguientes paquetes NUEVOS:

sysstat

```

0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 79 no actualizados.
Se necesita descargar 436 kB de archivos.
Se utilizarán 1.438 kB de espacio de disco adicional después de esta operación.
Des:1 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 sysstat arm64 12.2.0-2 [436 kB]
Descargados 436 kB en 1s (477 kB/s)
Preconfigurando paquetes ...
Seleccionando el paquete sysstat previamente no seleccionado.
(Leyendo la base de datos ... 125760 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../sysstat_12.2.0-2_arm64.deb ...
Desempaquetando sysstat (12.2.0-2) ...
Configurando sysstat (12.2.0-2) ...

Creating config file /etc/default/sysstat with new version
update-alternatives: utilizando /usr/bin/sar.sysstat para proveer /usr/bin/sar (sar) en modo automático
Created symlink /etc/systemd/system/multi-user.target.wants/sysstat.service →
/lib/systemd/system/sysstat.service.

Procesando disparadores para man-db (2.9.1-1) ...
Procesando disparadores para systemd (245.4-4ubuntu3.4) ...
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
 libjpeg-turbo8-dev
Se instalarán los siguientes paquetes NUEVOS:
 libjpeg-turbo8-dev libjpeg8-dev
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 79 no actualizados.
Se necesita descargar 216 kB de archivos.
Se utilizarán 623 kB de espacio de disco adicional después de esta operación.
Des:1 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 libjpeg-turbo8-dev arm64 2.0.3-
0ubuntu1.20.04.1 [215 kB]
Des:2 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 libjpeg8-dev arm64 8c-2ubuntu8 [1.550 B]
Descargados 216 kB en 1s (258 kB/s)
Seleccionando el paquete libjpeg-turbo8-dev:arm64 previamente no seleccionado.
(Leyendo la base de datos ... 125817 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../libjpeg-turbo8-dev_2.0.3-0ubuntu1.20.04.1_arm64.deb ...
Desempaquetando libjpeg-turbo8-dev:arm64 (2.0.3-0ubuntu1.20.04.1) ...
Seleccionando el paquete libjpeg8-dev:arm64 previamente no seleccionado.
Preparando para desempaquetar .../libjpeg8-dev_8c-2ubuntu8_arm64.deb ...
Desempaquetando libjpeg8-dev:arm64 (8c-2ubuntu8) ...
Configurando libjpeg-turbo8-dev:arm64 (2.0.3-0ubuntu1.20.04.1) ...
Configurando libjpeg8-dev:arm64 (8c-2ubuntu8) ...
Collecting testresources
  Downloading testresources-2.0.1-py2.py3-none-any.whl (36 kB)

```



```

> bufferutil@4.0.3 install /usr/local/lib/node_modules/bufferutil
> node-gyp-build
+ bufferutil@4.0.3
added 2 packages from 2 contributors in 6.161s

npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use
Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for
details.

/usr/local/bin/pm2 -> /usr/local/lib/node_modules/pm2/bin/pm2
/usr/local/bin/pm2-dev -> /usr/local/lib/node_modules/pm2/bin/pm2-dev
/usr/local/bin/pm2-docker -> /usr/local/lib/node_modules/pm2/bin/pm2-docker
/usr/local/bin/pm2-runtime -> /usr/local/lib/node_modules/pm2/bin/pm2-runtime

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@~2.3.2
(node_modules/pm2/node_modules/chokidar/node_modules/fsevents):

npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted
{"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"arm64"})

+ pm2@5.1.0
added 180 packages from 203 contributors in 35.433s
> utf-8-validate@5.0.5 install /home/odroid/.django-monitor/node_modules/utf-8-validate
> node-gyp-build

npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN package.json@1.0.0 No repository field.
+ utf-8-validate@5.0.5
added 2 packages from 2 contributors and audited 2 packages in 5.842s
found 0 vulnerabilities

> bufferutil@4.0.3 install /home/odroid/.django-monitor/node_modules/bufferutil
> node-gyp-build

npm WARN package.json@1.0.0 No repository field.
+ bufferutil@4.0.3
added 1 package from 1 contributor and audited 3 packages in 5.745s
found 0 vulnerabilities

npm WARN package.json@1.0.0 No repository field.
+ os@0.1.2
added 1 package from 1 contributor and audited 4 packages in 2.053s
found 0 vulnerabilities

npm WARN package.json@1.0.0 No repository field.
+ websocket@1.0.34
added 14 packages from 9 contributors and audited 18 packages in 5.755s
found 0 vulnerabilities

npm WARN package.json@1.0.0 No repository field.
+ http@0.0.1-security
added 1 package and audited 19 packages in 1.817s
found 0 vulnerabilities

npm WARN package.json@1.0.0 No repository field.
+ child_process@1.0.2

```

```

added 1 package and audited 20 packages in 1.832s
found 0 vulnerabilities

    •
    •
    •

[PM2] Init System found: systemd
Platform systemd
Template
[Unit]
Description=PM2 process manager
Documentation=https://pm2.keymetrics.io/
After=network.target
[Service]
Type=forking
User=odroid
LimitNOFILE=infinity
LimitNPROC=infinity
LimitCORE=infinity

Environment=PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/
bin:/usr/bin:/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin

Environment=PM2_HOME=/home/odroid/.pm2
PIDFile=/home/odroid/.pm2/pm2.pid
Restart=on-failure
ExecStart=/usr/local/lib/node_modules/pm2/bin/pm2 resurrect
ExecReload=/usr/local/lib/node_modules/pm2/bin/pm2 reload all
ExecStop=/usr/local/lib/node_modules/pm2/bin/pm2 kill
[Install]
WantedBy=multi-user.target
Target path
/etc/systemd/system/pm2-odroid.service
Command list
[ 'systemctl enable pm2-odroid' ]
[PM2] Writing init configuration in /etc/systemd/system/pm2-odroid.service
[PM2] Making script booting at startup...
[PM2] [-] Executing: systemctl enable pm2-odroid...
Created symlink /etc/systemd/system/multi-user.target.wants/pm2-odroid.service → /etc/systemd/system/pm2-
odroid.service.

[PM2] [v] Command successfully executed.
+-----+
[PM2] Freeze a process list on reboot via:
$ pm2 save
[PM2] Remove init script via:
$ pm2 unstartup systemd

```

```
[PM2] Saving current process list...
[PM2] Successfully saved in /home/odroid/.pm2/dump.pm2
-----INSTALATION FINISHED ENJOY-----
-----Master Configuration Finished! -----
```

6.1.1.17 Probar fecha y hora bien configuradas

Ejecutamos el comando `date` para cerciorarnos que la fecha y hora están bien configuradas según el contenido de `odroid_cluster.conf`, que la define como la hora de España.

```
root@odroid:~/odroid-cluster# date
Wed Jul 21 20:37:35 CEST 2021
```

6.1.2 Evaluación del slave

A continuación se muestra la salida de la inicialización de un nodo slave, comenzando por el script `add_slave.sh` y continuando con el script `init_slave.sh`, del cual puede verse su salida en el fichero `/var/log/odroid-cluster/init_slave_odroidX.out`.

6.1.2.1 Add_slave.sh

```
root@master:/home/odroid# cat /var/log/odroid_cluster/add_slave_odroid1.out

# odroid1:22 SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.1

/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"

/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed

/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@odroid1'"

and check to make sure that only the key(s) you wanted were added.

Configuring node odroid1. This may take a while. You can check the installation progress in
/var/log/odroid_cluster/init_slave_odroid1.out
```

Además, se puede comprobar que se está ejecutando el script `init_slave.sh` en los nodos si miramos el estado del servicio `dnsmasq`.

- `dnsmasq.service` - `dnsmasq` - A lightweight DHCP and caching DNS server
 - Loaded: loaded (/lib/systemd/system/dnsmasq.service; enabled; vendor preset: enabled)
 - Active: active (running) since Mon 2021-07-26 22:24:19 CEST; 10min ago
 - Main PID: 35125 (dnsmasq)

```

Memory: 4.3M
CGroup: /system.slice/dnsmasq.service
└─35125 /usr/sbin/dnsmasq -x /run/dnsmasq/dnsmasq.pid -u dnsmasq -7 /etc/dnsmasq.d,.dpkg-
dist,.dpkg-old,.dpkg-new --local-service --tru>
└─35126 /usr/sbin/dnsmasq -x /run/dnsmasq/dnsmasq.pid -u dnsmasq -7 /etc/dnsmasq.d,.dpkg-
dist,.dpkg-old,.dpkg-new --local-service --tru>
└─47760 /bin/bash /opt/scripts/add_slave.sh odroid1 0
└─47859 /bin/bash /opt/scripts/add_slave.sh odroid2 15
└─48118 ssh -t root@odroid2 ~/init_slave.sh 172.16.0.1 0 15 es_ES\;es\;cat\;Europe/Madrid
└─48119 ssh -t root@odroid1 ~/init_slave.sh 172.16.0.1 0 15 es_ES\;es\;cat\;Europe/Madrid

```

```

Jul 26 22:27:57 master sudo[47961]: pam_unix(sudo:session): session closed for user root
Jul 26 22:27:57 master sudo[47991]: pam_unix(sudo:session): session closed for user root
Jul 26 22:28:46 master dnsmasq-dhcp[35125]: DHCPDISCOVER(eth0) 192.168.2.238 00:1e:06:33:4d:fa
Jul 26 22:28:46 master dnsmasq-dhcp[35125]: DHCPOFFER(eth0) 172.16.0.231 00:1e:06:33:4d:fa
Jul 26 22:28:46 master dnsmasq-dhcp[35125]: DHCPREQUEST(eth0) 172.16.0.231 00:1e:06:33:4d:fa
Jul 26 22:28:46 master dnsmasq-dhcp[35125]: DHCPACK(eth0) 172.16.0.231 00:1e:06:33:4d:fa odroid2
Jul 26 22:28:46 master dnsmasq-dhcp[35125]: DHCPDISCOVER(eth0) 192.168.2.238 00:1e:06:33:ce:35
Jul 26 22:28:46 master dnsmasq-dhcp[35125]: DHCPOFFER(eth0) 172.16.0.41 00:1e:06:33:ce:35
Jul 26 22:28:46 master dnsmasq-dhcp[35125]: DHCPREQUEST(eth0) 172.16.0.41 00:1e:06:33:ce:35
Jul 26 22:28:46 master dnsmasq-dhcp[35125]: DHCPACK(eth0) 172.16.0.41 00:1e:06:33:ce:35 odroid1

```

6.1.2.2 Init_slave.sh

6.1.2.2.1 Arreglar un problema con el idioma

```

Generating locales (this might take a while)...
es_ES.UTF-8... done
Generation complete.
Generating locales (this might take a while)...
es_ES.UTF-8... done
Generation complete.

```

6.1.2.2.2 Establecer el máster como servidor DNS y DHCP

```

Internet Systems Consortium DHCP Client 4.4.1
Copyright 2004-2018 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Too few arguments.
Listening on LPF/eth0/00:1e:06:33:ce:35
Sending on LPF/eth0/00:1e:06:33:ce:35
Sending on Socket/fallback
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 3 (xid=0x55088153)
DHCPOFFER of 172.16.0.41 from 172.16.0.1
DHCPREQUEST for 172.16.0.41 on eth0 to 255.255.255.255 port 67 (xid=0x53810855)
DHCPACK of 172.16.0.41 from 172.16.0.1 (xid=0x55088153)

```

bound to 172.16.0.41 -- renewal in 18026 seconds.

6.1.2.2.3 Update

Get:1 http://archive.canonical.com/ubuntu focal InRelease [12.1 kB]

Get:2 http://ports.ubuntu.com/ubuntu-ports focal InRelease [265 kB]

.
.
.

Get:50 http://ports.ubuntu.com/ubuntu-ports focal-security/multiverse arm64 Packages [2472 B]

Get:51 http://ports.ubuntu.com/ubuntu-ports focal-security/multiverse Translation-en [4952 B]

Fetches 31.1 MB in 14s (2229 kB/s)

Reading package lists... Done

Reading package lists... Done

Building dependency tree

Reading state information... Done

Calculating upgrade... Done

The following packages have been kept back:

u-boot-tools ubuntu-advantage-tools

The following packages will be upgraded:

apt apt-utils base-files binutils binutils-aarch64-linux-gnu binutils-common
ca-certificates dirmngr distro-info-data dnsmasq-base gcc-10-base gnupg
gnupg-110n gnupg-utils gpg gpg-agent gpg-wks-client gpg-wks-server gpgconf
gpgsm gpgv initramfs-tools initramfs-tools-bin initramfs-tools-core
isc-dhcp-client isc-dhcp-common libapt-pkg6.0 libatomic1 libbinutils
libbluetooth3 libc-bin libc-dev-bin libc6 libc6-dev libcc1-0 libctf-nobfd0
libctf0 libcurl3-gnutls libgcc-s1 libgcc1 libglib2.0-0 libglib2.0-data
libgomp1 libhogweed5 libitm1 libldap-2.4-2 libldap-common liblsan0 liblz4-1
libnetplan0 libnettle7 libnss-systemd libpam-modules libpam-modules-bin
libpam-runtime libpam-systemd libpam0g libpolkit-agent-1-0
libpolkit-gobject-1-0 libprocps8 libpython3.8-minimal libpython3.8-stdlib
libseccomp2 libssl1.1 libstdc++6 libsystemd0 libtsan0 libubsan1 libudev1
libwbclient0 libx11-6 libx11-data libxml2 libzstd1 linux-base linux-firmware
linux-libc-dev locales lz4 netplan.io ntpdate openssh-client openssh-server
openssh-sftp-server openssl policykit-1 procps python-apt-common python3-apt
python3-update-manager python3-yaml python3.8 python3.8-minimal sudo systemd
systemd-sysv systemd-timesyncd tzdata ubuntu-keyring udev wpasupplicant

101 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.

Need to get 146 MB of archives.

After this operation, 18.2 MB of additional disk space will be used.

Get:1 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 libc6-dev arm64 2.31-0ubuntu9.2 [2058

kB]

.
.
.

Get:100 http://ports.ubuntu.com/ubuntu-ports focal-updates/universe arm64 ntpdate arm64 1:4.2.8p12+dfsg-3ubuntu4.20.04.1 [47.2 kB]

Get:101 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 wpasupplicant arm64 2:2.9-1ubuntu4.3 [1090 kB]

Fetched 146 MB in 16s (9351 kB/s)

Extracting templates from packages: 100%

Preconfiguring packages ...

(Reading database ... 39294 files and directories currently installed.)

Preparing to unpack .../0-libc6-dev_2.31-0ubuntu9.2_arm64.deb ...

Unpacking libc6-dev:arm64 (2.31-0ubuntu9.2) over (2.31-0ubuntu9.1) ...

•
•
•

Setting up isc-dhcp-client (4.4.1-2.1ubuntu5.20.04.2) ...

Setting up libctf-nobfd0:arm64 (2.34-6ubuntu1.1) ...

Setting up libnetplan0:arm64 (0.102-0ubuntu1~20.04.2) ...

Setting up python3-yaml (5.3.1-1ubuntu0.1) ...

Setting up dnsmasq-base (2.80-1.1ubuntu1.4) ...

Setting up libgomp1:arm64 (10.3.0-1ubuntu1~20.04) ...

Setting up locales (2.31-0ubuntu9.2) ...

Generating locales (this might take a while)...

es_ES.UTF-8... done

Generation complete.

Setting up libldap-common (2.4.49+dfsg-2ubuntu1.8) ...

Setting up libwbclient0:arm64 (2:4.11.6+dfsg-0ubuntu1.9) ...

Setting up ntpdate (1:4.2.8p12+dfsg-3ubuntu4.20.04.1) ...

Setting up tzdata (2021a-0ubuntu0.20.04) ...

6.1.2.2.4 Validar la hora y zona horaria

Current default time zone: 'Europe/Madrid'

Local time is now: Thu Jul 22 12:53:10 CEST 2021.

Universal Time is now: Thu Jul 22 10:53:10 UTC 2021.

Run 'dpkg-reconfigure tzdata' if you wish to change it.

6.1.2.2.5 Instalar sysstat

Reading package lists... Done

Building dependency tree

Reading state information... Done

The following additional packages will be installed:

libsensors-config libsensors5

Suggested packages:

lm-sensors isag

The following NEW packages will be installed:

libsensors-config libsensors5 sysstat

0 upgraded, 3 newly installed, 0 to remove and 2 not upgraded.

Need to get 468 kB of archives.

After this operation, 1579 kB of additional disk space will be used.

Get:1 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 libsensors-config all 1:3.6.0-2ubuntu1 [6092 B]

Get:2 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 libsensors5 arm64 1:3.6.0-2ubuntu1 [26.1 kB]

Get:3 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 sysstat arm64 12.2.0-2 [436 kB]

Fetches 468 kB in 1s (477 kB/s)

Preconfiguring packages ...

Selecting previously unselected package libsensors-config.

(Reading database ... 39320 files and directories currently installed.)

Preparing to unpack .../libsensors-config_1%3a3.6.0-2ubuntu1_all.deb ...

Unpacking libsensors-config (1:3.6.0-2ubuntu1) ...

Selecting previously unselected package libsensors5:arm64.

Preparing to unpack .../libsensors5_1%3a3.6.0-2ubuntu1_arm64.deb ...

Unpacking libsensors5:arm64 (1:3.6.0-2ubuntu1) ...

Selecting previously unselected package sysstat.

Preparing to unpack .../sysstat_12.2.0-2_arm64.deb ...

Unpacking sysstat (12.2.0-2) ...

Setting up libsensors-config (1:3.6.0-2ubuntu1) ...

Setting up libsensors5:arm64 (1:3.6.0-2ubuntu1) ...

Setting up sysstat (12.2.0-2) ...

Creating config file /etc/default/sysstat with new version

update-alternatives: using /usr/bin/sar.sysstat to provide /usr/bin/sar (sar) in auto mode

Created symlink /etc/systemd/system/multi-user.target.wants/sysstat.service →
/lib/systemd/system/sysstat.service.

Processing triggers for systemd (245.4-4ubuntu3.11) ...

Processing triggers for libc-bin (2.31-0ubuntu9.2) ...

6.1.2.2.6 Instalar y configurar NFS

Reading package lists... Done

Building dependency tree

Reading state information... Done

The following additional packages will be installed:

keyutils libevent-2.1-7 libnfsidmap2 libtirpc-common libtirpc3 rpcbind

Suggested packages:

open-iscsi watchdog

The following NEW packages will be installed:

keyutils libevent-2.1-7 libnfsidmap2 libtirpc-common libtirpc3 nfs-common
rpcbind

0 upgraded, 7 newly installed, 0 to remove and 2 not upgraded.

Need to get 516 kB of archives.

After this operation, 1812 kB of additional disk space will be used.

Get:1 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 libtirpc-common all 1.2.5-1 [7632 B]

Get:2 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 libtirpc3 arm64 1.2.5-1 [74.1 kB]

```

Get:3 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 rpcbind arm64 1.2.5-8 [41.1 kB]
Get:4 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 keyutils arm64 1.6-6ubuntu1 [43.6 kB]
Get:5 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 libevent-2.1-7 arm64 2.1.11-stable-1 [125 kB]
Get:6 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 libnfsidmap2 arm64 0.25-5.1ubuntu1 [27.1 kB]
Get:7 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 nfs-common arm64 1:1.3.4-2.5ubuntu3.4
[197 kB]
  Fetched 516 kB in 1s (556 kB/s)
  Selecting previously unselected package libtirpc-common.
  (Reading database ... 39389 files and directories currently installed.)
  Preparing to unpack .../0-libtirpc-common_1.2.5-1_all.deb ...
  Unpacking libtirpc-common (1.2.5-1) ...
  Selecting previously unselected package libtirpc3:arm64.
  Preparing to unpack .../1-libtirpc3_1.2.5-1_arm64.deb ...
  Unpacking libtirpc3:arm64 (1.2.5-1) ...
  Selecting previously unselected package rpcbind.
  Preparing to unpack .../2-rpcbind_1.2.5-8_arm64.deb ...
  Unpacking rpcbind (1.2.5-8) ...
  Selecting previously unselected package keyutils.
  Preparing to unpack .../3-keyutils_1.6-6ubuntu1_arm64.deb ...
  Unpacking keyutils (1.6-6ubuntu1) ...
  Selecting previously unselected package libevent-2.1-7:arm64.
  Preparing to unpack .../4-libevent-2.1-7_2.1.11-stable-1_arm64.deb ...
  Unpacking libevent-2.1-7:arm64 (2.1.11-stable-1) ...
  Selecting previously unselected package libnfsidmap2:arm64.
  Preparing to unpack .../5-libnfsidmap2_0.25-5.1ubuntu1_arm64.deb ...
  Unpacking libnfsidmap2:arm64 (0.25-5.1ubuntu1) ...
  Selecting previously unselected package nfs-common.
  Preparing to unpack .../6-nfs-common_1%3a1.3.4-2.5ubuntu3.4_arm64.deb ...
  Unpacking nfs-common (1:1.3.4-2.5ubuntu3.4) ...
  Setting up libtirpc-common (1.2.5-1) ...
  Setting up libevent-2.1-7:arm64 (2.1.11-stable-1) ...
  Setting up keyutils (1.6-6ubuntu1) ...
  Setting up libnfsidmap2:arm64 (0.25-5.1ubuntu1) ...
  Setting up libtirpc3:arm64 (1.2.5-1) ...
  Setting up rpcbind (1.2.5-8) ...
  Created symlink /etc/systemd/system/multi-user.target.wants/rpcbind.service →
/lib/systemd/system/rpcbind.service.
  Created symlink /etc/systemd/system/sockets.target.wants/rpcbind.socket → /lib/systemd/system/rpcbind.socket.
  Setting up nfs-common (1:1.3.4-2.5ubuntu3.4) ...

  Creating config file /etc/idmapd.conf with new version
  Adding system user `statd' (UID 109) ...
  Adding new user `statd' (UID 109) with group `nogroup' ...

```

```

Not creating home directory `~/var/lib/nfs'.
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-client.target → /lib/systemd/system/nfs-client.target.
Created symlink /etc/systemd/system/remote-fs.target.wants/nfs-client.target → /lib/systemd/system/nfs-client.target.
nfs-utils.service is a disabled or a static unit, not starting it.
Processing triggers for systemd (245.4-4ubuntu3.11) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...

```

Para validar que NFS se ha configurado correctamente, podemos mirar el directorio home del usuario odroid en el nodo slave y éste deberá contener lo mismo que en el directorio home de odroid en el máster.

```

odroid@master:/root/odroid-cluster$ ssh odroid1
Warning: Permanently added the ECDSA host key for IP address '172.16.0.41' to the list of known hosts.
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 3.16.85-65 aarch64)

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

```

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

```

odroid@odroid1:~$ ls
bin monitoring

```

6.1.2.2.7 Instalar Munge

```

Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libmunge2
The following NEW packages will be installed:
  libmunge2 munge
0 upgraded, 2 newly installed, 0 to remove and 2 not upgraded.
Need to get 105 kB of archives.
After this operation, 385 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 libmunge2 arm64 0.5.13-2build1 [19.4 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 munge arm64 0.5.13-2build1 [86.0 kB]

```

```

Fetched 105 kB in 1s (144 kB/s)
Selecting previously unselected package libmunge2.
(Reading database ... 39522 files and directories currently installed.)
Preparing to unpack .../libmunge2_0.5.13-2build1_arm64.deb ...
Unpacking libmunge2 (0.5.13-2build1) ...
Selecting previously unselected package munge.
Preparing to unpack .../munge_0.5.13-2build1_arm64.deb ...
Unpacking munge (0.5.13-2build1) ...
Setting up libmunge2 (0.5.13-2build1) ...
Setting up munge (0.5.13-2build1) ...
Generating a pseudo-random key using /dev/urandom completed.
Please refer to /usr/share/doc/munge/README.Debian for instructions to generate more secure key.
Created          symlink          /etc/systemd/system/multi-user.target.wants/munge.service      →
/lib/systemd/system/munge.service.
Processing triggers for systemd (245.4-4ubuntu3.11) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
Synchronizing state of munge.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable munge
2+0 records in
2+0 records out
1024 bytes (1.0 kB, 1.0 KiB) copied, 0.00243188 s, 421 kB/s

```

6.1.2.2.8 Instalar Slurm

Reading package lists... Done

Building dependency tree

Reading state information... Done

The following additional packages will be installed:

```

fontconfig fontconfig-config fonts-dejavu-core freeipmi-common libaec0
libcairo2 libcurl4 libdatrie1 libdbi1 libfontconfig1 libfreeipmi17
libfreetype6 libgfortran5 libgraphite2-3 libharfbuzz0b libhdf5-103
libhwloc-plugins libhwloc15 libipmimonitoring6 libltdl7 liblua5.1-0
libmysqlclient21 libnuma1 libpango-1.0-0 libpangocairo-1.0-0
libpangoft2-1.0-0 libpciaccess0 libpixmap-1-0 libpng16-16 librrd8 libs2
libthai-data libthai0 libxcb-render0 libxcb-shm0 libxnvctrl0 libxrender1
mysql-common ocl-icd-libopencl1 slurm-client slurm-wlm-basic-plugins
slurmctld slurmd

```

Suggested packages:

```

freeipmi-tools pciutils opencl-icd

```

The following NEW packages will be installed:

```

fontconfig fontconfig-config fonts-dejavu-core freeipmi-common libaec0
libcairo2 libcurl4 libdatrie1 libdbi1 libfontconfig1 libfreeipmi17
libfreetype6 libgfortran5 libgraphite2-3 libharfbuzz0b libhdf5-103
libhwloc-plugins libhwloc15 libipmimonitoring6 libltdl7 liblua5.1-0
libmysqlclient21 libnuma1 libpango-1.0-0 libpangocairo-1.0-0

```

```

libpangoft2-1.0-0 libpciaccess0 libpixmap-1-0 libpng16-16 librrd8 libsz2
libthai-data libthai0 libxcb-render0 libxcb-shm0 libxvctrl0 libxrender1
mysql-common ocl-icd-libopencl1 slurm-client slurm-wlm
slurm-wlm-basic-plugins slurmctld slurmd
0 upgraded, 44 newly installed, 0 to remove and 2 not upgraded.
Need to get 10.5 MB of archives.
After this operation, 40.4 MB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 freeipmi-common all 1.6.4-3ubuntu1.1
[179 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 libnuma1 arm64 2.0.12-1 [20.5 kB]
      .
      .
      .
Get:43 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 ocl-icd-libopencl1 arm64 2.2.11-1ubuntu1 [29.2
kB]
Get:44 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 libhwloc-plugins arm64 2.1.0+dfsg-4 [13.8
kB]
Fetched 10.5 MB in 2s (4273 kB/s)
Extracting templates from packages: 100%
Selecting previously unselected package freeipmi-common.
(Reading database ... 39560 files and directories currently installed.)
Preparing to unpack ../00-freeipmi-common_1.6.4-3ubuntu1.1_all.deb ...
Unpacking freeipmi-common (1.6.4-3ubuntu1.1) ...
      .
      .
      .
Setting up slurm-wlm-basic-plugins (19.05.5-1) ...
Setting up slurmd (19.05.5-1) ...
update-alternatives: using /usr/sbin/slurmd-wlm to provide /usr/sbin/slurmd (slurmd) in auto mode
Created symlink /etc/systemd/system/multi-user.target.wants/slurmd.service →
/lib/systemd/system/slurmd.service.
Setting up slurm-client (19.05.5-1) ...
Setting up slurmctld (19.05.5-1) ...
update-alternatives: using /usr/sbin/slurmctld-wlm to provide /usr/sbin/slurmctld (slurmctld) in auto mode
Created symlink /etc/systemd/system/multi-user.target.wants/slurmctld.service →
/lib/systemd/system/slurmctld.service.
Setting up slurm-wlm (19.05.5-1) ...
Processing triggers for systemd (245.4-4ubuntu3.11) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
Synchronizing state of slurmd.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable slurmd
Created symlink /etc/systemd/system/multi-user.target.wants/slurm_watcher.path →
/etc/systemd/system/slurm_watcher.path.
Created symlink /etc/systemd/system/multi-user.target.wants/slurm_watcher.service →
/etc/systemd/system/slurm_watcher.service.

```

6.1.2.2.9 Instalar Ganglia

Reading package lists... Done

Building dependency tree

Reading state information... Done

wget is already the newest version (1.20.3-1ubuntu1).

wget set to manually installed.

The following additional packages will be installed:

```
apache2-bin apache2-data apache2-utils galera-3 gawk libaio1
libapache2-mod-php7.4 libapr1 libaprutil1 libaprutil1-dbd-sqlite3
libaprutil1-ldap libcgi-fast-perl libcgi-pm-perl libconfig-inifiles-perl
libdbd-mysql-perl libdbi-perl libencode-locale-perl libfcgi-perl libgd3
libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
libhttp-date-perl libhttp-message-perl libio-html-perl libjbig0
libjpeg-turbo8 libjpeg8 liblua5.2-0 liblwp-mediatypes-perl libonig5
libreadline5 libsigsegv2 libsnappy1v5 libsodium23 libterm-readkey-perl
libtiff5 libtimedate-perl liburi-perl libwebp6 libxpm4 libzip5 lsof
mariadb-client-10.3 mariadb-client-core-10.3 mariadb-common
mariadb-server-10.3 mariadb-server-core-10.3 php-common php7.4 php7.4-cli
php7.4-common php7.4-curl php7.4-gd php7.4-json php7.4-mbstring php7.4-mysql
php7.4-openssl php7.4-readline php7.4-zip psmisc rsync socat ssl-cert
```

Suggested packages:

```
apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser ufw
gawk-doc php-pear libclone-perl libmldbm-perl libnet-daemon-perl
libsql-statement-perl libgd-tools libdata-dump-perl libipc-sharedcache-perl
libwww-perl mailx mariadb-test tinyca openssl-blacklist zip
```

The following NEW packages will be installed:

```
apache2 apache2-bin apache2-data apache2-utils galera-3 gawk libaio1
libapache2-mod-php libapache2-mod-php7.4 libapr1 libaprutil1
libaprutil1-dbd-sqlite3 libaprutil1-ldap libcgi-fast-perl libcgi-pm-perl
libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl libencode-locale-perl
libfcgi-perl libgd3 libhtml-parser-perl libhtml-tagset-perl
libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl
libjbig0 libjpeg-turbo8 libjpeg8 liblua5.2-0 liblwp-mediatypes-perl libonig5
libreadline5 libsigsegv2 libsnappy1v5 libsodium23 libterm-readkey-perl
libtiff5 libtimedate-perl liburi-perl libwebp6 libxpm4 libzip5 lsof
mariadb-client-10.3 mariadb-client-core-10.3 mariadb-common mariadb-server
mariadb-server-10.3 mariadb-server-core-10.3 php php-common php-curl php-gd
php-mbstring php-mysql php-zip php7.4 php7.4-cli php7.4-common php7.4-curl
php7.4-gd php7.4-json php7.4-mbstring php7.4-mysql php7.4-openssl
php7.4-readline php7.4-zip psmisc rsync socat ssl-cert unzip
```

0 upgraded, 74 newly installed, 0 to remove and 2 not upgraded.

Need to get 26.9 MB of archives.

After this operation, 197 MB of additional disk space will be used.

```

Get:1 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 rsync arm64 3.1.3-8 [308 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 libapr1 arm64 1.6.5-1ubuntu1 [85.4 kB]
      .
      .
      .

Get:74 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 unzip arm64 6.0-25ubuntu1 [163 kB]
Fetched 26.9 MB in 4s (6389 kB/s)
Extracting templates from packages: 100%
Preconfiguring packages ...
Selecting previously unselected package rsync.
(Reading database ... 40194 files and directories currently installed.)
Preparing to unpack .../00-rsync_3.1.3-8_arm64.deb ...
Unpacking rsync (3.1.3-8) ...
      .
      .
      .

Setting up rsync (3.1.3-8) ...
Created symlink /etc/systemd/system/multi-user.target.wants/rsync.service → /lib/systemd/system/rsync.service.
Setting up libaprutil1:arm64 (1.6.1-4ubuntu2) ...
Setting up libjpeg8:arm64 (8c-2ubuntu8) ...
Setting up php-common (2:75) ...
Created symlink /etc/systemd/system/timers.target.wants/phpsessionclean.timer →
/lib/systemd/system/phpsessionclean.timer.
Setting up libhttp-date-perl (6.05-1) ...
Setting up php7.4-common (7.4.3-4ubuntu2.5) ...
Creating config file /etc/php/7.4/mods-available/calendar.ini with new version
Creating config file /etc/php/7.4/mods-available/ctype.ini with new version
Creating config file /etc/php/7.4/mods-available/exif.ini with new version
      .
      .
      .

Enabling module mpm_prefork.
info: Executing deferred 'a2enmod php7.4' for package libapache2-mod-php7.4
Enabling module php7.4.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service →
/lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service →
/lib/systemd/system/apache-htcacheclean.service.

Setting up mariadb-server (1:10.3.29-0ubuntu0.20.04.1) ...
Setting up php7.4 (7.4.3-4ubuntu2.5) ...
Setting up libapache2-mod-php (2:7.4+75) ...
Setting up php-gd (2:7.4+75) ...
Setting up php (2:7.4+75) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...

```

```

Processing triggers for systemd (245.4-4ubuntu3.11) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for php7.4-cli (7.4.3-4ubuntu2.5) ...
Processing triggers for libapache2-mod-php7.4 (7.4.3-4ubuntu2.5) ...
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libconfuse-common libconfuse2 libganglia1
Suggested packages:
  librds-perl
The following NEW packages will be installed:
  ganglia-monitor libconfuse-common libconfuse2 libganglia1 rrdtool
0 upgraded, 5 newly installed, 0 to remove and 2 not upgraded.
Need to get 521 kB of archives.
After this operation, 1849 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 libconfuse-common all 3.2.2+dfsg-1 [5512 B]
Get:2 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 libconfuse2 arm64 3.2.2+dfsg-1 [22.6 kB]
Get:3 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 libganglia1 arm64 3.6.0-7ubuntu4 [54.2 kB]
Get:4 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 ganglia-monitor arm64 3.6.0-7ubuntu4 [67.2
kB]
Get:5 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 rrdtool arm64 1.7.2-3build2 [371 kB]
Fetched 521 kB in 1s (529 kB/s)
Selecting previously unselected package libconfuse-common.
(Reading database ... 42240 files and directories currently installed.)
Preparing to unpack ../libconfuse-common_3.2.2+dfsg-1_all.deb ...
Unpacking libconfuse-common (3.2.2+dfsg-1) ...
Selecting previously unselected package libconfuse2:arm64.
Preparing to unpack ../libconfuse2_3.2.2+dfsg-1_arm64.deb ...
Unpacking libconfuse2:arm64 (3.2.2+dfsg-1) ...
Selecting previously unselected package libganglia1.
Preparing to unpack ../libganglia1_3.6.0-7ubuntu4_arm64.deb ...
Unpacking libganglia1 (3.6.0-7ubuntu4) ...
Selecting previously unselected package ganglia-monitor.
Preparing to unpack ../ganglia-monitor_3.6.0-7ubuntu4_arm64.deb ...
Unpacking ganglia-monitor (3.6.0-7ubuntu4) ...
Selecting previously unselected package rrdtool.
Preparing to unpack ../rrdtool_1.7.2-3build2_arm64.deb ...
Unpacking rrdtool (1.7.2-3build2) ...
Setting up libconfuse-common (3.2.2+dfsg-1) ...
Setting up libconfuse2:arm64 (3.2.2+dfsg-1) ...
Setting up rrdtool (1.7.2-3build2) ...
Setting up libganglia1 (3.6.0-7ubuntu4) ...

```

```
Setting up ganglia-monitor (3.6.0-7ubuntu4) ...
Adding system user: ganglia.
Processing triggers for systemd (245.4-4ubuntu3.11) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
```

6.1.2.2.10 Instalar Mpich

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  gfortran gfortran-9 hwloc-nox libgfortran-9-dev libmpich-dev libmpich12
Suggested packages:
  gfortran-doc gfortran-9-doc libcoarrays-dev mpich-doc
The following NEW packages will be installed:
  gfortran gfortran-9 hwloc-nox libgfortran-9-dev libmpich-dev libmpich12
  mpich
0 upgraded, 7 newly installed, 0 to remove and 2 not upgraded.
Need to get 9912 kB of archives.
After this operation, 42.0 MB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 libgfortran-9-dev arm64 9.3.0-17ubuntu1~20.04 [390 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 gfortran-9 arm64 9.3.0-17ubuntu1~20.04 [6373 kB]
Get:3 http://ports.ubuntu.com/ubuntu-ports focal/main arm64 gfortran arm64 4:9.3.0-1ubuntu2 [1360 B]
Get:4 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 hwloc-nox arm64 2.1.0+dfsg-4 [145 kB]
Get:5 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 libmpich12 arm64 3.3.2-2build1 [1005 kB]
Get:6 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 mpich arm64 3.3.2-2build1 [345 kB]
Get:7 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 libmpich-dev arm64 3.3.2-2build1 [1654 kB]
Fetched 9912 kB in 2s (4273 kB/s)
Selecting previously unselected package libgfortran-9-dev:arm64.
(Reading database ... 42372 files and directories currently installed.)
Preparing to unpack .../0-libgfortran-9-dev_9.3.0-17ubuntu1~20.04_arm64.deb ...
Unpacking libgfortran-9-dev:arm64 (9.3.0-17ubuntu1~20.04) ...
Selecting previously unselected package gfortran-9.
Preparing to unpack .../1-gfortran-9_9.3.0-17ubuntu1~20.04_arm64.deb ...
Unpacking gfortran-9 (9.3.0-17ubuntu1~20.04) ...
Selecting previously unselected package gfortran.
Preparing to unpack .../2-gfortran_4%3a9.3.0-1ubuntu2_arm64.deb ...
Unpacking gfortran (4:9.3.0-1ubuntu2) ...
Selecting previously unselected package hwloc-nox.
Preparing to unpack .../3-hwloc-nox_2.1.0+dfsg-4_arm64.deb ...
Unpacking hwloc-nox (2.1.0+dfsg-4) ...
Selecting previously unselected package libmpich12:arm64.
Preparing to unpack .../4-libmpich12_3.3.2-2build1_arm64.deb ...
```

```

Unpacking libmpich12:arm64 (3.3.2-2build1) ...
Selecting previously unselected package mpich.
Preparing to unpack .../5-mpich_3.3.2-2build1_arm64.deb ...
Unpacking mpich (3.3.2-2build1) ...
Selecting previously unselected package libmpich-dev:arm64.
Preparing to unpack .../6-libmpich-dev_3.3.2-2build1_arm64.deb ...
Unpacking libmpich-dev:arm64 (3.3.2-2build1) ...
Setting up hwloc-nox (2.1.0+dfsg-4) ...
Setting up libgfortran-9-dev:arm64 (9.3.0-17ubuntu1~20.04) ...
Setting up libmpich12:arm64 (3.3.2-2build1) ...
Setting up gfortran-9 (9.3.0-17ubuntu1~20.04) ...
Setting up mpich (3.3.2-2build1) ...
update-alternatives: using /usr/bin/mpicc.mpich to provide /usr/bin/mpicc (mpi) in auto mode
update-alternatives: using /usr/bin/mpirun.mpich to provide /usr/bin/mpirun (mpirun) in auto mode
Setting up gfortran (4:9.3.0-1ubuntu2) ...
update-alternatives: using /usr/bin/gfortran to provide /usr/bin/f95 (f95) in auto mode
update-alternatives: using /usr/bin/gfortran to provide /usr/bin/f77 (f77) in auto mode
Setting up libmpich-dev:arm64 (3.3.2-2build1) ...
update-alternatives: using /usr/include/aarch64-linux-gnu/mpich to provide /usr/include/aarch64-linux-gnu/mpi
(mpi-aarch64-linux-gnu) in auto mode
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...

```

6.1.2.2.11 Upgrade

Por motivos de espacio solo incluyo el final de la salida de esta acción:

```

Procesando disparadores para libgdk-pixbuf2.0-0:arm64 (2.40.0+dfsg-3ubuntu0.2) ...
Procesando disparadores para libc-bin (2.31-0ubuntu9.2) ...
Procesando disparadores para dbus (1.12.16-2ubuntu2.1) ...
Procesando disparadores para systemd (245.4-4ubuntu3.11) ...
Procesando disparadores para rygel (0.38.3-1ubuntu1) ...
Procesando disparadores para sgml-base (1.29.1) ...
Connection to odroid1 closed.

```

6.1.3 Resultado final

Una vez ha finalizado la inicialización de los nodos, podemos verificar que hemos obtenido un sistema paralelo distribuido. Con el comando `sinfo`, verificamos que ambos nodos se han configurado correctamente, y de la misma manera se ha configurado el Slurm y el NFS, ya que el fichero de configuración de Slurm se comparte a todos los nodos mediante NFS para que todos tengan la misma configuración, lo cual es un requisito de Slurm.

```

root@master:~# sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
main*      up       infinite     2    idle odroid[1-2]
root@master:~# █

```

Figura 5. Prueba de la correcta configuración de Slurm.

A continuación, compilamos algunos de los programas implementados con MPI para probar la configuración del sistema.

```
root@master:~/odroid-cluster/mpi_scripts# mpicc kmean_mpi.c -O3
root@master:~/odroid-cluster/mpi_scripts# mpicc mD5K_mpi.c -O3
root@master:~/odroid-cluster/mpi_scripts# █
```

Figura 6. Compilación de programas para la evaluación del sistema.

Este proceso nos genera un ejecutable, que por defecto y si no especificamos otro nombre, se llamará *a.out*. A continuación se muestran una serie de ejecuciones del programa correspondiente al código *mD5K_mpi.c*, variando el número de procesos que lo ejecutan:

```
odroid@master:~$ time srun -n2 a.out

Numero elements de la matriu dispersa C 9840962 proces : 0
Suma dels elements de C 31844954756

real    2m23,900s
user    0m0,010s
sys     0m0,020s
odroid@master:~$ time srun -n4 a.out
srun: error: odroid1: task 3: Killed
```

Figura 7. Ejecución del programa *mD5K_mpi.c* con 2 y 4 threads.

Como se observa en la imagen, la primera ejecución con 2 procesos se ha completado de forma correcta; sin embargo, la segunda, con 4 procesos, ha arrojado un error. En el apartado 6.2.2, en la figura 13, muestro los gráficos de la ejecución de estos programas para demostrar y verificar el funcionamiento de Ganglia. Las métricas recogidas por la herramienta me hacen pensar que este código, el *mD5K_mpi.c*, requiere de unas capacidades que ahora mismo, con 2 nodos slave, mi sistema no puede ofrecer.

Es por ello por lo que a continuación pruebo la ejecución del código *kmean_mpi.c*.

```

odroid@master:~$ time srun -n2 a.out
(2496) Theta; -54.825, 27.418 cost: 549.549

real    3m4,538s
user    0m0,020s
sys     0m0,020s
odroid@master:~$ time srun -n4 a.out
(2496) Theta; -54.825, 27.418 cost: 549.549

real    1m54,294s
user    0m0,000s
sys     0m0,030s
odroid@master:~$ time srun -n8 a.out
(2496) Theta; -54.825, 27.418 cost: 549.549

real    1m3,181s
user    0m0,020s
sys     0m0,030s
odroid@master:~$ █

```

Figura 8. Ejecución del programa *kmean_mpi.c* con 2, 4 y 8 threads.

En esta ocasión mi sistema sí cuenta con capacidad para hacer frente a esta ejecución, y además se comprueba la correcta ejecución en paralelo del código, ya que la ejecución con 2, 4 y 8 procesos han arrojado el mismo resultado, pero con tiempos de ejecución menores en cuanto al incremento en el número de procesos.

De esta manera, se considera probada la correcta configuración del clúster, cumpliendo con el objetivo de automatizar la configuración de un sistema paralelo distribuido.

6.2 Evaluación de los sistemas de monitorización

6.2.1 Ganglia

Ganglia es la herramienta de monitorización escogida para la versión final del sistema. Ésta se instala y configura automáticamente en el proceso de inicialización del clúster. Para visualizar las métricas, debemos abrir un navegador como Midori desde la placa máster, y una vez abierto teclear en el navegador la dirección <http://localhost/ganglia>. Al hacerlo, visualizamos la página inicial de Ganglia.

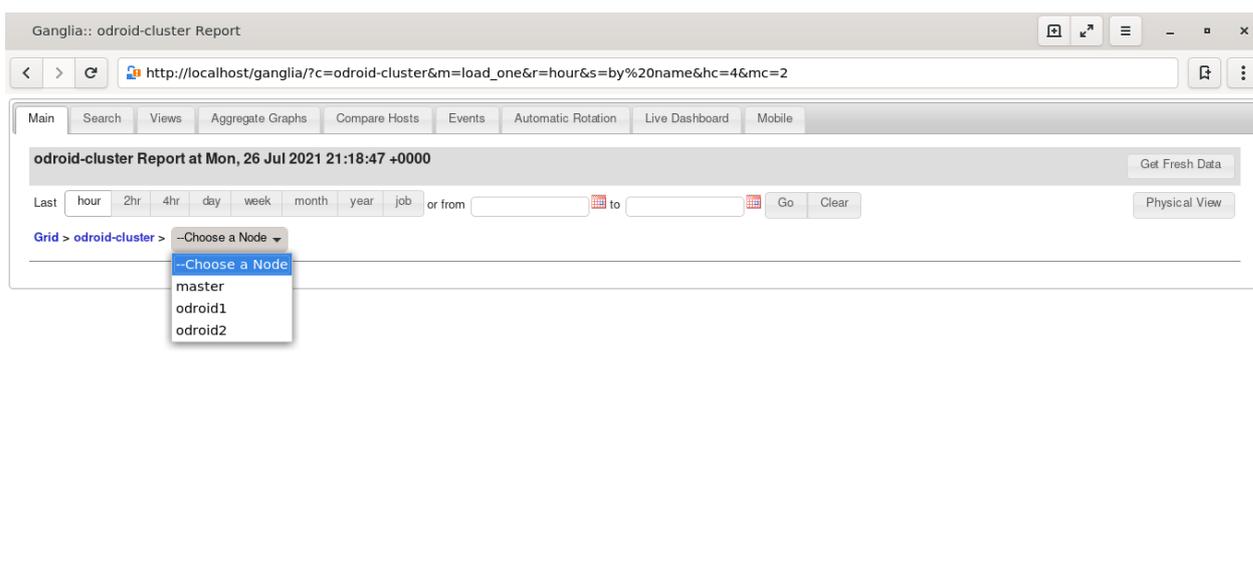


Figura 9. Página inicial de la herramienta Ganglia

En ella, podemos visualizar las distintas funcionalidades que nos ofrece esta herramienta en las pestañas superiores, como son la visualización de los gráficos, agregarlos para ofrecer una vista distinta, comparar hosts, programar eventos, etc.

También podemos observar en el desplegable la lista de nodos de nuestro clúster, tanto el nodo máster como los slaves añadidos.

Si seleccionamos por ejemplo el nodo máster, podemos echar un vistazo a los gráficos que genera de las métricas que el servicio *ganglia-monitor* instalado en el máster recoge y envía a la página web.

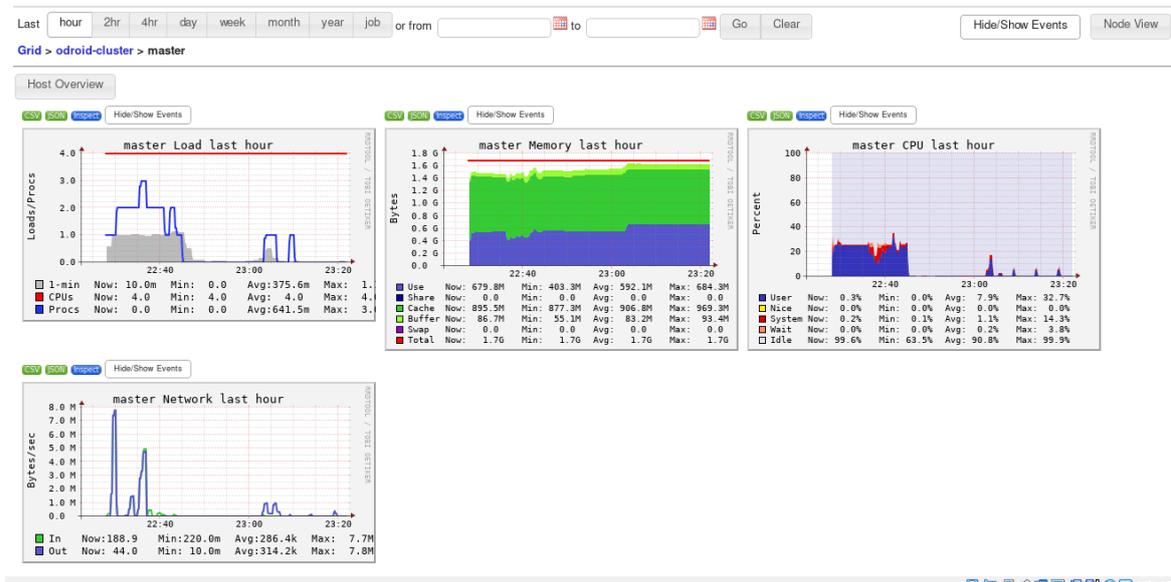


Figura 10. Gráficos de las métricas del nodo máster en la página de Ganglia (1).

Estas imágenes están tomadas poco después de hacer una instalación desde el inicio del sistema, por lo que podemos apreciar la carga de trabajo que ha soportado el sistema durante este proceso. En el primer gráfico, apreciamos la carga de procesos del máster, llegando al pico de 3 cuando se estaban ejecutando al mismo tiempo tanto el script *init_master.sh* como los dos *init_slave.sh*. En cuanto a la memoria RAM de la máquina,

vemos como está usándose toda, aunque parte de ésta está compartida y otra parte es caché. Por otro lado, vemos la actividad de la CPU durante el proceso de instalación, que no es muy elevado, presentando un pico de algo más del 30%, por lo que considero que en cuanto a potencia de CPU está correcto. Por último, podemos ver el estado de la red en cuanto a la cantidad de información que ha pasado por el máster. El pico que se observa en el gráfico corresponde con el momento de actualización de los nodos slaves durante su inicialización.

Además de estos gráficos de métricas más principales o más empleadas, Ganglia muestra muchas más métricas, como las que vemos en la figura 11, por ejemplo.



Figura 11. Gráficos de las métricas del nodo máster en la página de Ganglia (2).

En cuanto a los nodos slaves, en el momento de tomar estas imágenes vemos una situación muy distinta, ya que no han tenido mucho trabajo hasta ese momento.

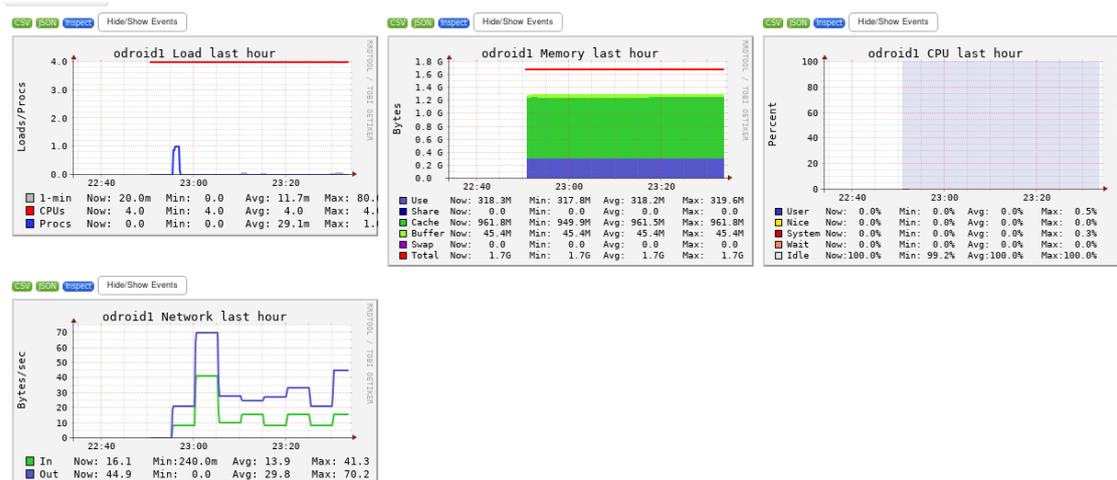


Figura 12. Gráficos de las métricas del nodo slave 'odroid1' en la página de Ganglia en el momento de inicializar el nodo.

Podemos observar que no hay una gran cantidad de trabajo excepto por la cantidad de información tanto de salida como de entrada que se observa en su actividad de red. Cabe destacar que en el gráfico que muestra el porcentaje de CPU utilizado no se muestra

nada ya que al momento de inicializar el slave, el servicio *ganglia-monitor* no se inició correctamente y no recogió esas métricas. No me ha parecido relevante volverlo a probar ya que, como vemos en el máster, éste no presenta mucha actividad de CPU y el problema por el que no se inició ya está corregido y probado que no ocurre.

Para constatar la efectividad de la herramienta de monitorización y para dar fiabilidad al sistema, aquí muestro los gráficos recogidos durante la ejecución de las pruebas con *srun* en el apartado 6.1.3.

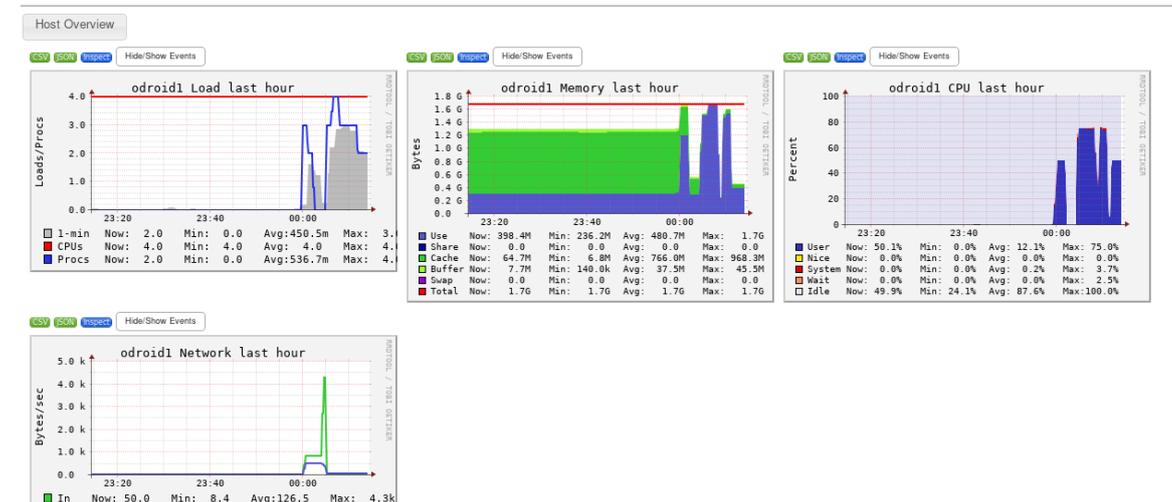


Figura 13. Gráficos de las métricas del nodo slave 'odroid1' tras la ejecución del código *mD5K_mpi.c*.

Mirando las métricas del nodo, podemos observar un pico de procesos y de memoria, acompañado de un alto porcentaje de CPU utilizado. Estos datos me hacen pensar que el programa que he ejecutado requiere de demasiada potencia y/o capacidad de memoria, unos requerimientos que el sistema con 2 placas slave no puede soportar.

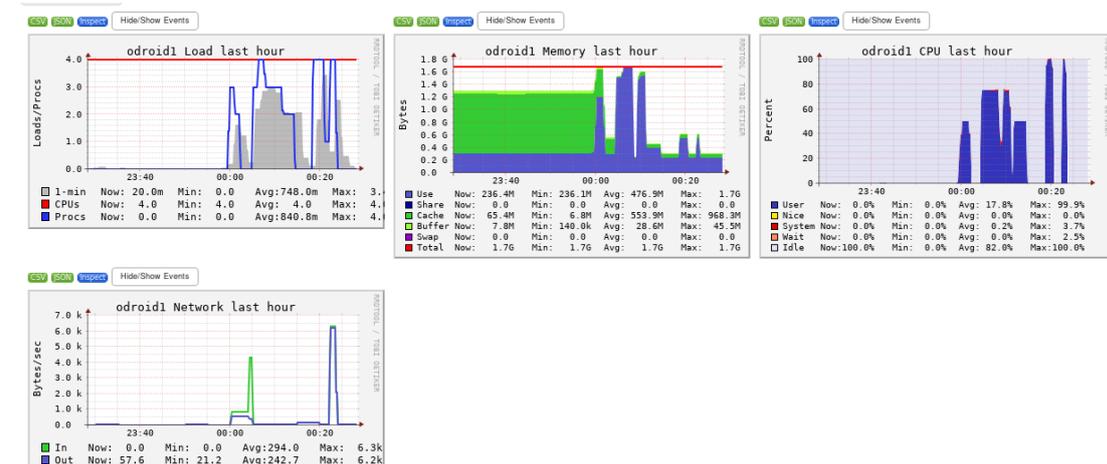


Figura 14. Gráficos de las métricas del nodo slave 'odroid1' tras la ejecución del código *kmean_mpi.c*.

En esta última imagen, la figura 14, podemos observar las métricas recogidas en la ejecución del código *kmean_mpi.c*, correspondiente a la figura 8, que valida la correcta configuración del sistema y verifica la funcionalidad de la herramienta Ganglia como herramienta de monitorización. En ella podemos ver, entre otras cosas, los 4 procesos que se han ejecutado en la placa, los máximos de los que dispone, además de un porcentaje de

uso de la CPU del 100%, verificando que nuestro sistema es capaz de ofrecer todos sus recursos para la ejecución en paralelo.

6.2.2 Monitorix

Para poder evaluar la herramienta Monitorix, primero debemos establecer que reconozca todos los nodos del clúster. Debido a las complicaciones por automatizar la instalación de Monitorix en todos los nodos y ya que esta herramienta no va a estar en la versión definitiva del sistema y solo la incluyo aquí a modo de experimentación, la instalación de Monitorix se ha hecho mediante scripts pero su configuración se ha hecho manualmente en el momento del test. Con la configuración inicial, solo podemos ver las métricas del localhost, que en este caso es el Odroid máster. Por ello, en el fichero `/etc/monitorix/monitorix.conf` debemos incluir las IPs del resto de nodos del clúster.

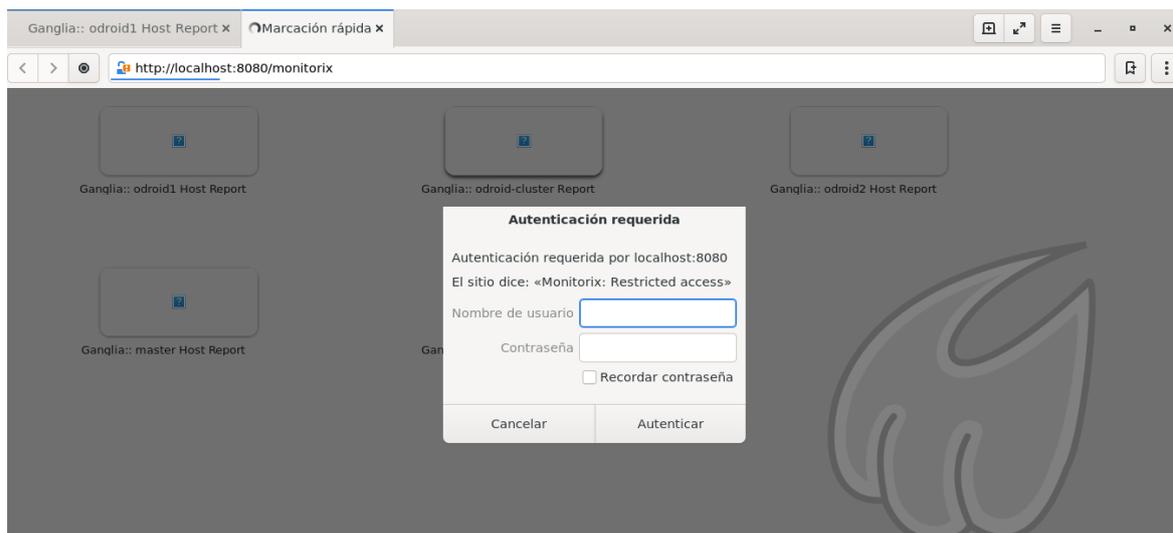


Figura 15. Visión de la ventana de autenticación al conectar con la página de Monitorix.

Una vez correctamente configurado, abrimos el navegador desde la placa máster y ponemos en la barra de navegación la dirección `http://localhost:8080/monitorix`. Lo primero que aparece al acceder a la página web es un recuadro que pide que te identifiques mediante usuario y contraseña. Esto aparece debido a la configuración del `htpasswd` para permitir el acceso mediante contraseña. En este caso, se ha establecido como usuario por defecto 'odroid' mientras que la contraseña la escribirá el usuario en el momento de la instalación. Puede modificarse o crear nuevos usuarios en el archivo `/var/lib/monitorix/htpasswd`.

Una vez se introduce el usuario y contraseña, aparece la página principal de Monitorix.

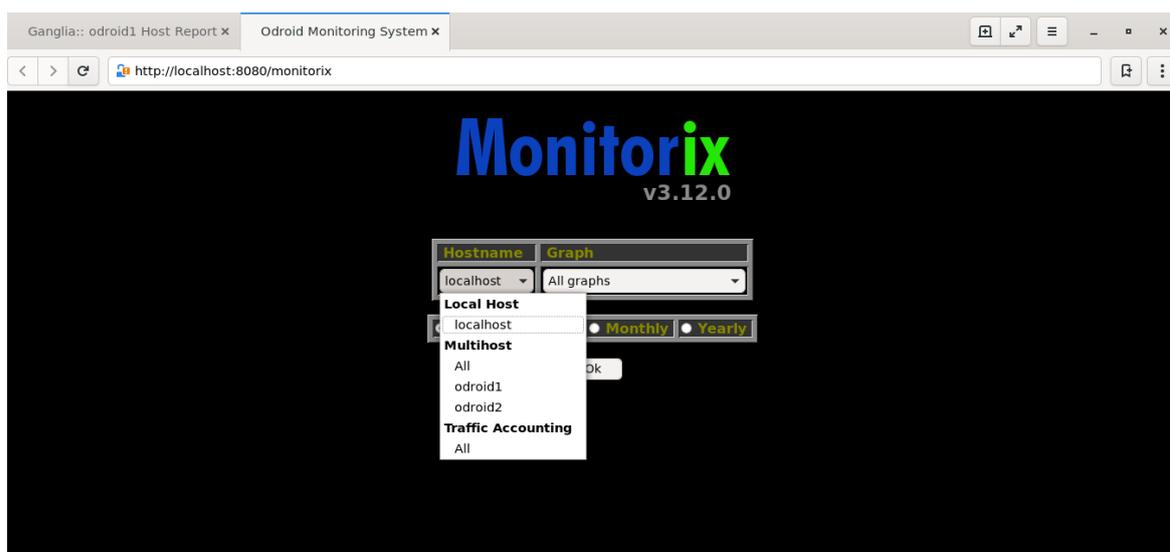


Figura 16. Página principal de la herramienta Monitorix.

En el desplegable, podemos observar la máquina localhost, que es el nodo máster, y en el apartado multihost visualizamos los nodos slave que tenemos instalados en el sistema. En el apartado graph podemos seleccionar los gráficos que deseamos visualizar. Para mostrar todo lo que ofrece esta herramienta, dejaremos la opción de *All graphs*.

Al seleccionar las opciones localhost y *All graphs*, clicamos en Ok para visualizar los gráficos. Como podemos observar en la figura 17, se nos muestran gran variedad de gráficos con las métricas del nodo máster. Sin embargo, únicamente tenemos la opción de visualizar estos gráficos, no podemos hacer nada más con ellos, y ésta ha sido una de las razones por las que finalmente me he decantado por Ganglia como herramienta de monitorización.



Figura 17. Gráficos del nodo máster desde la página de Monitorix (1).



Figura 18. Gráficos del nodo máster desde la página de Monitorix (2).

Para visualizar las métricas, he ejecutado los programas de MPI al igual que con Ganglia, para así verificar el funcionamiento de Monitorix y dejar constancia de que es una herramienta de monitorización igualmente válida y que podría implementarse. Actualmente es una herramienta en desarrollo, por lo que en un futuro se tendrá que tener en cuenta, ya que también está muy bien.

6.3 Evaluación del sistema de administración remota

Por último, quería demostrar la funcionalidad de los sistemas de administración remota para la ejecución de comandos en los distintos nodos del clúster, tanto utilizando la interfaz de la página web interna del sistema como mediante la ejecución de los comandos ubicados en el directorio */bin* del directorio *home* del usuario por defecto.

6.3.1 Vía interfaz de la página web interna

Para comprobar la funcionalidad del sistema de administración remota mediante la interfaz que se encuentra en la página web interna, he conectado con todas las placas del clúster y he ejecutado unos comandos básicos para que se pueda comprobar la funcionalidad.

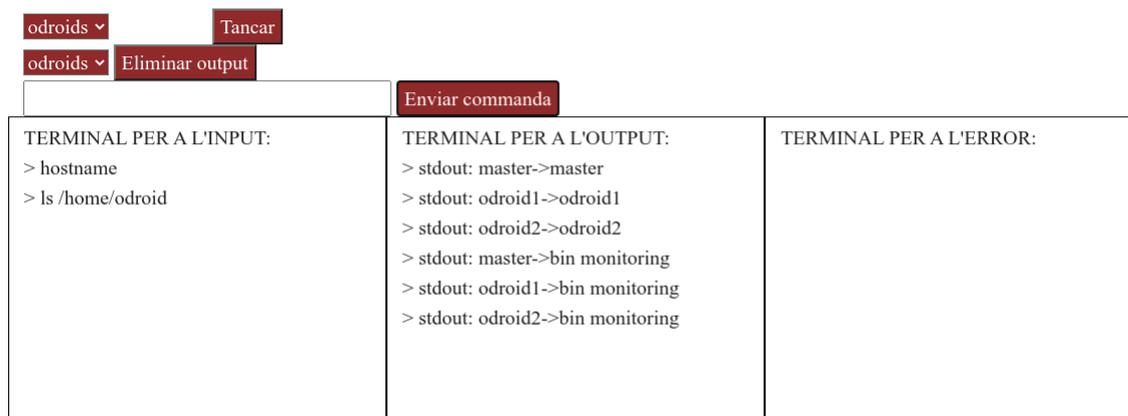


Figura 19. Prueba del sistema de administración remota mediante la interfaz web (1).

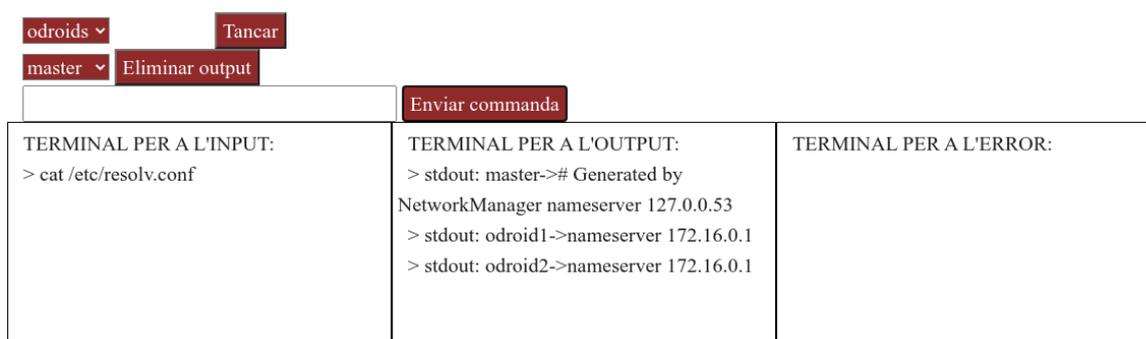


Figura 20. Prueba del sistema de administración remota mediante la interfaz web (2).

En las dos imágenes se puede apreciar el diseño de la interfaz. Ésta nos permite seleccionar a qué placas nos queremos conectar y a partir de ahí ejecutar comandos que se envían a todas ellas. Podemos observar en la primera imagen la salida del comando `hostname`, en la que cada placa responde con su nombre, por lo que se ejecuta de manera correcta. También se observa la salida del comando `ls` sobre el directorio `/home/odroid` de cada nodo, y al ser esta salida la misma para todas las placas, se confirma también que la compartición del directorio `home` del usuario `odroid` mediante NFS funciona correctamente. Por último, en la segunda imagen, vemos la salida del contenido del archivo `/etc/resolv.conf` para constatar que es diferente en el máster respecto a los slaves y confirmar que se ejecuta en todos los nodos.

6.3.2 Vía scripts ubicados en `/home/odroid/bin/`

Para probar este sistema de administración remota mediante scripts, he realizado las mismas pruebas ejecutando el script `global_execute_par`. El inconveniente actualmente de este método es que pide las contraseñas del usuario `root` para ejecutar, pero como beneficio te permite ejecutar comandos con permisos de administrador, cosa que desde la interfaz web no se puede hacer.

```

root@odroid:~/bin# ./global_execute_par hostname
executo sudo un cop y aixi serveix pels altres
Linux master 3.16.85-65 #1 SMP PREEMPT Mon Jan 18 13:32:38 UTC 2021 aarch64 aarc
h64 aarch64 GNU/Linux
Executo: hostname
root@odroid:~/bin# Pseudo-terminal will not be allocated because stdin is not a
terminal.
Pseudo-terminal will not be allocated because stdin is not a terminal.
odroid1
odroid2

```

Figura 21. Prueba de la ejecución del script de administración remota en paralelo (1).

```

root@odroid:~/bin# ./global_execute_seq hostname
executo sudo un cop y aixi serveix pels altres
Linux master 3.16.85-65 #1 SMP PREEMPT Mon Jan 18 13:32:38 UTC 2021 aarch64 aarc
h64 aarch64 GNU/Linux
Executo: hostname
odroid1
Connection to odroid1 closed.
odroid2
Connection to odroid2 closed.

```

Figura 22. Prueba de la ejecución del script de administración remota en secuencial (1).

```

root@master:~/bin# ./global_execute_seq cat /etc/resolv.conf
executo sudo un cop y aixi serveix pels altres
Linux master 3.16.85-65 #1 SMP PREEMPT Mon Jan 18 13:32:38 UTC 2021 aarch64 aarc
h64 aarch64 GNU/Linux
Executo: cat /etc/resolv.conf
nameserver 172.16.0.1
Connection to odroid1 closed.
nameserver 172.16.0.1
Connection to odroid2 closed.
root@master:~/bin#

```

Figura 23. Prueba de la ejecución del script de administración remota en secuencial (2).

```

root@master:~/bin# ./global_execute_par cat /etc/resolv.conf
executo sudo un cop y aixi serveix pels altres
Linux master 3.16.85-65 #1 SMP PREEMPT Mon Jan 18 13:32:38 UTC 2021 aarch64 aarc
h64 aarch64 GNU/Linux
Executo: cat /etc/resolv.conf
root@master:~/bin# Pseudo-terminal will not be allocated because stdin is not a
terminal.
Pseudo-terminal will not be allocated because stdin is not a terminal.
nameserver 172.16.0.1
nameserver 172.16.0.1

```

Figura 24. Prueba de la ejecución del script de administración remota en paralelo (2).

7 Previsión de futuro

Éste es un proyecto que tiene mucho futuro, ya que está pensado para ser utilizado en entornos docentes, principalmente, como soporte en asignaturas que requieran de la computación en paralelo. Por lo tanto, y pese a que se ha logrado tener un sistema funcional y automatizado, considero que será necesario un seguimiento del proyecto, más o menos exhaustivo, ya que con el paso del tiempo surgen nuevas versiones de los sistemas operativos y de las herramientas, que pueden ocasionar conflictos de dependencias en el sistema y provocar que algo o todo deje de funcionar. Esto lo sé porque mi trabajo ha consistido en eso, en adaptar lo que estaba hecho para que sea funcional.

Por supuesto, como en todo sistema, existen mejoras que se podrían realizar. La mayoría de interacciones con el sistema, pese a contar éste con una interfaz gráfica, se han de realizar mediante consola de comandos y trabajando con los archivos de configuración de los principales servicios como dnsmasq, slurm o Ganglia. Esto implica que tengas que estar familiarizado con entornos Linux y más específicamente con los paquetes instalados y dónde se ubican estos archivos de configuración, cómo reiniciar servicios, etc. Además, la parte de administración remota está poco desarrollada, ya que la interfaz de administración por medio de la web interna está muy limitada y no permite la ejecución de comandos que necesiten permisos de administración o el redireccionamiento de la salida de un comando.

Estos aspectos serían los principales en cuanto a capacidad y potencial de mejora. Sin embargo, cabe destacar que el sistema es completamente funcional y cumple satisfactoriamente con las necesidades que tiene un sistema paralelo distribuido.

8 Conclusiones

El trabajo que he desarrollado supone parte de un proyecto que llevan los profesores que me han ofrecido su ayuda y sus consejos a la hora de realizar mi TFG. Es por ello por lo que este trabajo no es un trabajo más de tantos en la carrera, sino un proyecto real para ofrecer este sistema, este multicomputador de bajo coste, para que cualquiera que necesite un sistema de estas características, ya sea en un entorno docente o particular, pueda disponer de herramientas y conocimientos para montarlo y llevarlo a cabo. Es por todo esto que para mí este trabajo suponía algo más y he querido poner todo mi esfuerzo en conseguir algo funcional y lo más óptimo posible, algo que cuando yo deje este proyecto pueda continuar y funcionar para ofrecer este servicio que considero clave sobre todo para la educación en informática y computación en paralelo.

Me gustaría recalcar que, pese a que este trabajo no lo he iniciado yo desde cero, ya que compañeros como David o Joan ya trabajaron en él, y antes que ellos estuvieron los profesores que iniciaron esto, a raíz de tener que juntar ambos trabajos y adaptarlos para que funcionara no en un entorno simulado como son las condiciones en las que he desarrollado el trabajo sino en el sistema real, he tenido que realizar mucho trabajo de inspección, de comprensión y de entendimiento de cómo funcionaba todo, de cómo interacciona cada script con el conjunto y testear una y otra vez detectando lo que no funcionaba y buscándole solución.

A nivel personal, este trabajo me ha supuesto un reto, que me ha permitido adentrarme mucho más en los sistemas Linux, en la administración de sistemas, sobretodo de entornos distribuidos. Ésta es una formación que, pese a que en parte se enseña durante el grado, es cierto que cuando te tienes que enfrentar a un problema real se hace mucho más laborioso el encontrar la manera correcta de hacerlo y conseguir que todo funcione como esperas. Junto a ello, al utilizar la monitorización del sistema como parte del trabajo, esto me ha llevado a investigar y reflexionar sobre aspectos más concretos a tener en cuenta a la hora de montar un sistema, como es tener en cuenta el consumo de recursos o el estado de la red. Esto, en un entorno docente como es la universidad, siempre está más controlado, pero cuando el sistema lo montas tú y está pensado para ser utilizado por cualquiera, debes poner más detalle en todos los aspectos para intentar que todo funcione como se espera y surjan los menos errores posibles.

Por último, me gustaría agradecer a los tres profesores que me han ayudado en esta tarea, Carles Aliagas, Carlos Molina y Pere Millán, por las facilidades que me han puesto y la ayuda prestada, ya que de no ser así no sé si hubiera podido sacar adelante este proyecto y seguro que no de la forma en la que está. Estoy muy orgulloso del trabajo que he realizado y espero que pueda servir para que el sistema se pueda utilizar y generar así ese soporte al aprendizaje que es para lo que está pensado.

9 Anexos

9.1 Listado de versiones

Listado de versiones de los paquetes con los que se ha verificado la instalación y configuración del clúster. Para conocer la versión de cada paquete, he utilizado el comando `apt-cache policy name`, siendo 'name' el nombre de la herramienta o paquete a consultar.

Tabla 1. Listado de versiones de paquetes utilizados en la configuración del sistema.

Paquete	Versión	Paquete	Versión
Ubuntu	Minimal 20.04	Gdm3	3.36.3-0ubuntu0.20.04.4
Dnsmasq	2.80-1.1ubuntu1.4	Xfce4	4.14
Expect	5.45.4-2build1	Xfce4-goodies	4.14.0
Net-tools	1.60+git20180626.aebd88e-1ubuntu1	Tightvncserver	1.3.10-0ubuntu5
Ifupdown	0.8.35ubuntu1	Munge	0.5.13-2build1
Nfs-common	1.3.4-2.5ubuntu3.4	Tar	1.30+dfsg-7ubuntu0.20.04.1
Nfs-kernel-server	1.3.4-2.5ubuntu3.4	Apt	2.0.2ubuntu0.2
Openssh-server	8.2p1-4ubuntu0.2	Systemd	245.4-4ubuntu3.4
Sshpass	1.06-1	Iptables-persistent	1.0.14
Midori	7.0-2.1	Netfilter-persistent	1.0.14
Mpich	3.3.2-2build1	Adduser	3.118ubuntu2
Slurm-wlm	19.05.5-1	Sysstat	12.2.0-2
Apache2	2.4.41-4ubuntu3.4	Rrdtool	1.7.2-3build2
Php	2:7.4+75	Wget	1.20.3-1ubuntu1
Mariadb-server	10.3.29-0ubuntu0.20.04.1	Gunicorn	20.0.4-3
Ganglia-monitor	3.6.0-7ubuntu4	Python3-django	2:2.2.12-1ubuntu0.7
Gmetad	3.6.0-7ubuntu4	Nginx	1.18.0-0ubuntu1.2
Ganglia-webfrontend	3.6.1-3	Nodejs	10.19.0~dfsg-3ubuntu1

9.2 Manual de Instalación

A continuación adjunto los manuales que he desarrollado, que contienen toda la información necesaria para instalar y configurar el sistema, y que se encuentran además disponibles en la web del proyecto (<http://cloudlab.urv.cat/minimulticomputer>).

9.2.1 Paso 1: Cargar Imágenes

Si es la primera vez que vas a usar las placas Odroid C2, necesitarás cargar un Sistema Operativo en ellas. Este proyecto ha sido desarrollado utilizando la imagen de Ubuntu Minimal 20.04, el cual es un Sistema Operativo de software libre y código abierto. En esta guía detallamos los pasos que debes seguir para cargar la imagen de Ubuntu en todas tus placas y así conseguir un **MiniMultiComputador de Bajo Coste Basado en Odroids**.

Si tus placas ya tienen la imagen de Ubuntu, pasa directamente al Paso 2: Montar *Hardware*.

Cargar Imagen en Todas las Placas Odroid C2

1. Dirígete a la página de wiki.odroid.com y descarga la versión de Ubuntu Minimal 20.04 para las placas Odroid C2.
2. Carga la imagen del Sistema Operativo Ubuntu Minimal en una placa (ver Opción 1 y Opción 2).
3. Repite el paso 2 para cada una de las placas.

Cargar Imagen en Una Placa Odroid C2 (Opción 1)

Esta opción funciona independientemente del Sistema Operativo que uses en tu ordenador.

1. Descarga e instala un programa para escribir imágenes de un Sistema Operativo. Te recomendamos BalenaEtcher (<https://www.balena.io/etcher/>).
2. Inserta la memoria flash eMMC de la placa Odroid C2 en tu ordenador. Necesitarás un lector adecuado para leer esa memoria.
3. Abre el programa BalenaEtcher.
4. Selecciona la imagen del Sistema Operativo que hemos descargado previamente.
5. Selecciona la tarjeta de memoria flash que acabas de conectar.
6. Haz click en Flash! y el programa cargará automáticamente la imagen en la memoria.
7. Desconecta la memoria flash y vuelve a ponerla en la placa Odroid C2.

Cargar Imagen en Una Placa Odroid C2 (Opción 2)

Esta opción funciona si tu ordenador utiliza un Sistema Operativo MacOS o Linux.

1. Una vez hayas conectado el lector y la tarjeta a tu PC, abre un terminal.
2. Puedes mirar el nombre de la partición mediante el comando `fdisk -l`. Fíjate en el resultado que aparece, ya que es importante no confundirse de partición o podrías borrar tu sistema.
3. La partición que corresponde a la memoria eMMC seguramente sea la única que tenga un tamaño igual que tu tarjeta de memoria (normalmente será de 8 GB). Para asegurarte, puedes mirar el resultado del comando `fdisk -l` tanto sin la memoria conectada como cuando sí lo esté. Aquella partición que aparezca nueva será tu tarjeta.
4. Puede ser necesario desmontar primero la partición. Esto se realiza con el comando `umount` “nombre de la partición”, donde el nombre de la partición será más o menos así: `/dev/sdX`, siendo X una letra o número.
5. Para cargar la imagen, deberás ejecutar: `dd bs=1M conv=fsync if=XXX.img of=/dev/sdX`, donde `if` (input file) será el nombre de la imagen que te has descargado, y `of` (output file) el nombre de la partición correspondiente a tu memoria y que has consultado con el comando `fdisk -l`.
6. Recomendamos no ejecutar este último paso con el símbolo `&` al final, para así garantizar que termina la escritura de forma adecuada. Además, se recomienda ejecutar el comando `sync` una vez que la escritura haya terminado.

En cuanto tengas instalada la imagen de Ubuntu en todas tus placas, puedes continuar con el *Paso 2: Montar Hardware* o si tienes ya la estructura montada, ve directamente al *Paso 3: Instalar Software*.

9.2.2 Paso 2: Montar Hardware

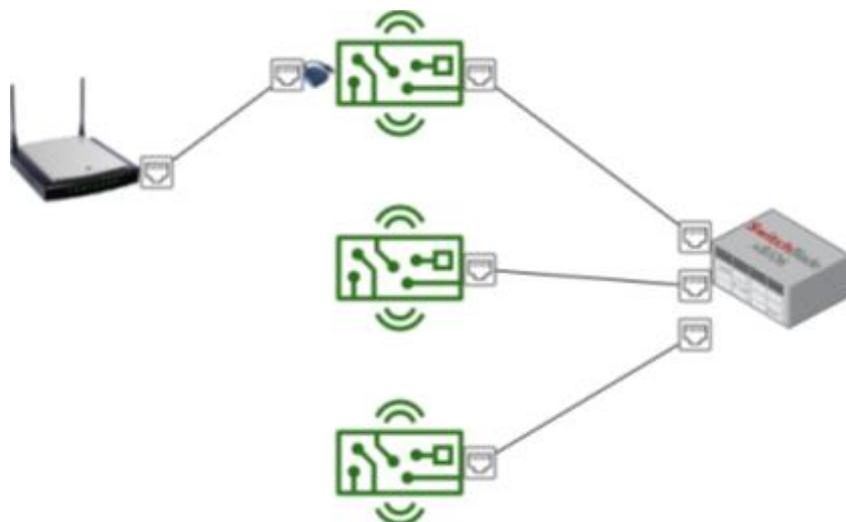
En este paso se indica el hardware requerido y las instrucciones necesarias para el montaje del **MiniMulticomputador de Bajo Coste Basado en Odroids**.

Listado de Hardware Requerido

- N placas Odroid C2 junto con su fuente de alimentación.
- 1 Switch con tantos puertos como placas Odroid C2, o varios switches.
- Tantos cables ethernet como placas Odroid C2, más 1 cable para conectar la placa máster con el router.
- 1 adaptador USB-Ethernet.
- 1 enchufe USB con tantos puertos como placas Odroid C2, o varios enchufes USB.
- 1 router con conexión a internet.
- 1 estructura o soporte para el montaje de las placas.
- 1 cable HDMI (opcional).

Instrucciones de Montaje

1. Montar las placas Odroid C2 sobre una estructura que permita la ventilación.
2. Conectar los cables ethernet al puerto ethernet respectivo de cada placa por un extremo y conectar el otro extremo a los puertos ethernet del Switch.
3. Conectar el adaptador USB-Ethernet en el puerto USB de la placa que hará la función de máster.
4. Conectar el cable ethernet restante al adaptador USB-Ethernet que acabamos de conectar y por el otro extremo conectarlo al router.
5. Si las placas no poseen una imagen Ubuntu (en el caso de que las acabes de comprar), ve al *Paso 1: Cargar Imágenes*.
6. Una vez todas las placas tengan la imagen de Ubuntu, conecta los cables de alimentación a la corriente utilizando el enchufe multi-USB. Conecta también el Switch a la corriente eléctrica.
7. Una vez terminados estos pasos, ya tienes el sistema hardware montado y listo para funcionar. En la figura siguiente puedes ver un esquema de la red resultante.



Ahora, para configurar el sistema, puedes ir al *Paso 3: Instalar Software*.

9.2.3 Paso 3: Instalar Software

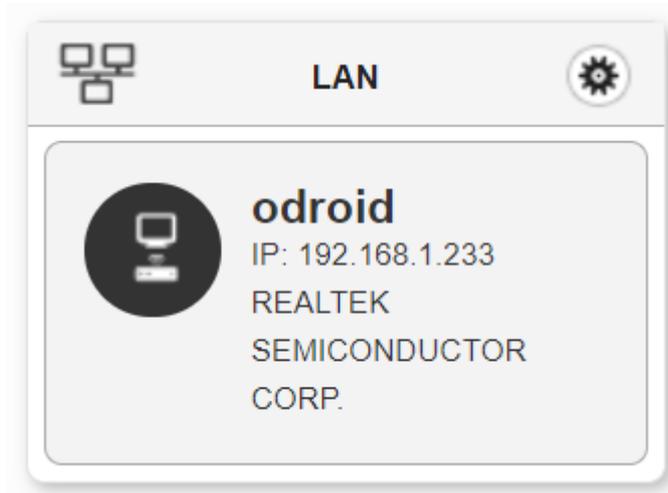
En este paso se indica cómo instalar el software necesario en los nodos (1 Máster y N Slaves) del **MiniMulticomputador de Bajo Coste Basado en Odroids** para que sea totalmente funcional. Este software se encuentra en el fichero comprimido (tgz) que encontrarás en el apartado de Descargas de la página principal. Descárgalo y guárdalo en un directorio conocido.

Configurar el Nodo Máster del MiniMultiComputador

1. Para comenzar la instalación, debemos conocer la dirección IP del nodo máster. Para ello, deberemos entrar en la página del router al que está conectado. Con un ordenador conectado a la misma red (WiFi o

por cable), abriremos un navegador y pondremos una de las siguientes direcciones IP: 192.168.0.1 o 192.168.1.1. Primero prueba con una dirección, y si no obtienes resultados, prueba con la otra, ya que dependiendo del proveedor de internet puede ubicarse en una dirección u otra. El usuario y contraseña para entrar en la página viene escrita en la parte inferior del router. Como última opción, realiza una búsqueda en Internet sobre cómo acceder al router de tu compañía en particular.

- Una vez en la página del router, se debe buscar la dirección IP de la placa Odroid que hará de máster, la cual aparecerá de un modo similar al que se muestra a continuación. Se debe tener en cuenta que cada router es diferente y puede ser más o menos fácil encontrarlo. Si no consigues encontrar la dirección IP de la placa Odroid, no te preocupes, pasa al siguiente punto en el que se explica una posible solución.



- Para instalar el software, debemos pasar el fichero comprimido que has descargado en el primer punto de este apartado a la placa que hará la función de máster. Para ello, debes conocer la IP de esta placa. Si has podido encontrar su dirección IP puedes pasar al último punto. Si por el contrario no has podido, aquí te proponemos una solución:
 - Ejecuta el comando `arp -v`.
 - Si no lo tienes instalado, se encuentra dentro del paquete *net-tools*.
 - Ejecuta `sudo apt install net-tools` y una vez instalado, ejecuta `arp -v`. Este comando mostrará los dispositivos conectados a la red. Tu ordenador se debe encontrar conectado al mismo router que la placa máster, y de esta manera te aparecerá su IP.
 - Si aparece más de una IP, puedes probar a ejecutar un `ssh root@direcciónIP`, sustituyendo *direcciónIP* por las direcciones que te aparezcan. Cuando conectes con la dirección de la placa odroid, te pedirá una contraseña (odroid) y al escribirla te dará acceso a la placa. De esta manera ya sabrás la dirección. Recuérdala para futuras conexiones.
- Si ya conoces la IP del máster, entonces estás listo para la instalación. Sitúate en el directorio donde tengas el fichero comprimido con el software, abre un terminal y ejecuta esta serie de comandos:
 - `ssh-keygen -R direcciónIP`
 - `scp odroid-cluster.tgz root@direcciónIP:/root`
 - `ssh root@direcciónIP`
 - `tar xzvf odroid-cluster.tgz`
 - `cd odroid-cluster`
 - `./init_master.sh`
- Tras estos comandos, comenzará la instalación y configuración automáticamente. Durante el transcurso de ella se pedirá una mínima interacción, como puede ser especificar las contraseñas de los usuarios root y odroid.
- Una vez finalizada la configuración, tendrás la placa máster lista para funcionar.

Añadir y Configurar los Nodos Slaves del MiniMultiComputador

- Para añadir un nodo al MiniMulticomputador, deberá situarse en el directorio `/root/odroid-cluster` como usuario root, y ejecutar el comando `./add_slave.sh odroidX`, donde X se sustituirá por el número de placas que haya instalado en el sistema más 1. Es decir, si ya tienes 2 nodos slaves configurados y quieres añadir un tercero, el comando será `./add_slave.sh odroid3`.
- Durante la instalación se pedirán las contraseñas del usuario odroid y del usuario root del nuevo nodo, la cual por defecto es 'odroid'.
- Tras esto, el nodo se configurará automáticamente. Puedes consultar el registro de instalación en el archivo `/var/log/odroid-cluster/init_slave_odroidX.out`.

9.2.4 Paso 4: Guía de Uso

Una vez que ya cuentas con la placa máster inicializada y uno o más nodos slave configurados, podrás comenzar a utilizar el clúster. Y esto incluye funciones como ejecutar programas en paralelo, visualizar las métricas de los nodos y el estado de la red, o administrar el sistema de manera remota desde el máster.

Ejecución en el clúster

El sistema cuenta con la herramienta Slurm, que proporciona un sistema de colas tanto para usuarios como para procesos para gestionar la ejecución en el clúster. A continuación explicamos algunas de las funcionalidades que esta herramienta ofrece:

- Puedes consultar el estado del clúster mediante el comando `sinfo`. Este comando te muestra el número de nodos con los que cuentas y el estado de los mismos. Si el estado es 'idle', es que el nodo se encuentra operativo. Si está en 'down', significa que algo ha fallado y no puedes utilizar ese nodo. Deberás revisarlo para ver si es problema de configuración o si se ha quedado colgado.
- Para ejecutar programas, utiliza el comando `srun`. Este te permite añadir opciones, como especificar el número de procesos que se crearán, el número de nodos a utilizar, redireccionar el resultado, etc.
- Cuando lances un programa a ejecutar, podrás consultar su estado, el tiempo de ejecución que lleva, su identificador (PID) y quién lo lanzó mediante el comando `squeue`.
- Si deseas cancelar la ejecución de un proceso, utiliza el comando `scancel` seguido del identificador (PID) del proceso a cancelar.

Para más información sobre la herramienta Slurm y cómo utilizarla, puedes consultar su documentación en la página web: <https://slurm.schedmd.com/documentation.html>.

Monitorización del sistema

El sistema cuenta con una herramienta de monitorización que se instala y configura durante la inicialización del sistema para que puedas hacer un seguimiento exhaustivo del clúster. Esta herramienta es Ganglia, y para acceder a la página web que muestra las métricas de tu clúster deberás acceder a un navegador desde la placa máster.

- El primer paso que debes hacer es conectarte a la placa máster. Si lo haces mediante SSH, utiliza la opción `-Y` en tu comando, para permitir el paso de la interfaz gráfica y puedas visualizar el navegador. Un ejemplo de conexión sería `ssh -Y odroid@192.168.X.X`, especificando la dirección IP de tu placa máster.
- Una vez te encuentres en la placa máster, abre un navegador web. El sistema cuenta con el navegador Midori instalado.
- Una vez lo hayas abierto, pon la siguiente dirección en la barra del navegador: `http://localhost/ganglia`.
- Se te mostrará la página principal de Ganglia. En la parte central-izquierda se muestra un desplegable, y al clicar en él te aparecerán todos los nodos de tu clúster. Para visualizar los datos de un nodo, no tienes más que clicar en su nombre y la página mostrará todos los gráficos que ha generado para dicho nodo.

Si deseas conocer más sobre las posibilidades que te ofrece esta herramienta, puedes echar un vistazo en su página web: <http://ganglia.sourceforge.net/>.

Administración remota

El sistema cuenta con un par de scripts para la administración y mantenimiento del clúster. Estos scripts permiten la ejecución de comandos en todos los nodos del clúster de manera remota desde la placa máster. De esta manera, se ofrece la posibilidad de consultar el estado de servicios o de ejecutar comandos de actualización o reinicio, por ejemplo, de una manera rápida y centralizada, sin necesidad de tener que conectarse uno a uno a todos los nodos del clúster.

Para esta tarea, se cuenta con dos scripts, llamados `global_execute_seq` y `global_execute_par`, que ejecutan el comando o secuencia de comandos con pipes que se les indique por parámetro en todos los nodos del clúster.

Aspectos a tener en cuenta:

- La lista de nodos del clúster para la ejecución de estos scripts viene recogida en el archivo `/opt/scripts/odroid.par`. Este archivo se modifica automáticamente tras la inicialización de cada nodo slave, para incluirlo. Si se desea excluir uno o más nodos de la ejecución de estos scripts de administración remota,

se debe comentar la línea correspondiente al nodo en cuestión del archivo `/opt/scripts/odroid.par`, es decir, habrá que incluir un '#' al inicio de la línea correspondiente.

- La ejecución de estos scripts, al encontrarse éstos bajo el directorio `bin/`, se puede hacer desde cualquier directorio. Un ejemplo de uso de estos sería el siguiente: `global_execute_seq hostname`, el cual arrojará como resultado el nombre asignado que tienen cada uno de los nodos slave.
- La diferencia entre ambos scripts es que el que cuenta con el sufijo `seq` ejecuta el comando de manera secuencial, es decir, espera a que termine la ejecución del comando en el primer nodo slave para lanzar la ejecución del siguiente slave. Por otro lado, el que cuenta con el sufijo `par` lanza todas las ejecuciones sin esperar a que ninguna previa termine. El primero, el secuencial, se deberá utilizar para tareas que requieran de paso de paquetes, como por ejemplo `apt update`, ya que como el nodo máster ejerce de cortafuegos de la red interna, una ejecución de este comando en paralelo conlleva la sobrecarga del máster. Para los comandos que no impliquen paso de paquetes a través del máster, se puede utilizar la opción en paralelo del script.

10 Referencias

1. C. Aliagas, M. García-Famoso, R. Meseguer, P. Millán and C. Molina, "A Low-Cost Multicomputer for Teaching Environments," in *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, vol. 15, no. 3, pp. 171-182, Aug. 2020, doi: 10.1109/RITA.2020.3008098.
2. Odroid Wiki. *OS Installation Guide*.
https://wiki.odroid.com/getting_started/os_installation_guide#tab_odroid-c2 [visitado el 27/06/2021].
3. Server World. Monitorix: *Enable Multi-Host*.
https://www.server-world.info/en/note?os=CentOS_8&p=monitorix&f=2 [visitado el 02/07/2021].
4. Atlantic.Net Blog. *How yo install Ganglia Monitoring Server on Ubuntu 18.04*.
<https://www.atlantic.net/vps-hosting/how-to-install-ganglia-monitoring-server-on-ubuntu-18-04/>. [visitado el 02/07/2021].