

David Ferrer Fernández

**AUTOMATITZACIÓ DE LA CREACIÓ D'UN SERVIDOR PARAL·LEL
DISTRIBUÏT BASAT EN ODOIDS**

TREBALL DE FI DE GRAU

dirigit per Carles Aliagas Castell

Grau d'Enginyeria Informàtica



UNIVERSITAT ROVIRA I VIRGILI

Tarragona

2020

Resum.

Aquest projecte s'ha desenvolupat per a l'aprenentatge i recerca de coneixements personals, a banda de tindre com a objectiu principal facilitar als usuaris, amb pocs coneixements sobre sistemes Linux i xarxes, poder muntar el seu propi servidor paral·lel distribuït, amb objectius acadèmics o de recerca, i pressupost ajustat. Al començament, es va fer un anàlisi de les necessitats del projecte. Un cop conegudes, es van anar seleccionant per separat, provant el seu funcionament i configuracions. Un cop complerts els requisits establerts, es va procedir a automatitzar-les mitjançant scripts. Per poder fer les proves en un entorn real i proper al que es trobarà l'usuari final es va muntar un clúster a petita escala per poder fer les proves pertinents. Els resultats obtinguts han estat majoritàriament exitosos, permetent una correcta automatització en el muntatge de tot el clúster, amb la mínima intervenció de l'usuari.

Resumen.

Este proyecto se ha desarrollado para el aprendizaje y búsqueda de conocimientos personales, además de tener como objetivo principal facilitar a los usuarios, con pocos conocimientos sobre sistemas Linux y redes, poder montar su propio servidor paralelo distribuido, con objetivos académicos o de investigación, y presupuesto ajustado. Al principio, se hizo un análisis de las necesidades del proyecto. Una vez conocidas, se fueron seleccionando por separado, probando su funcionamiento y configuraciones. Una vez cumplidos los requisitos establecidos, se procedió a automatizarse mediante scripts. Para poder hacer las pruebas en un entorno real y cercano al que se encontrará el usuario final se montó un clúster a pequeña escala para poder hacer las pruebas pertinentes. Los resultados obtenidos han sido mayoritariamente exitosos, permitiendo una correcta automatización en el montaje de todo el clúster, con la mínima intervención del usuario.

Abstract.

This project has been developed for learning and seeking personal knowledge, in addition to having as its main objective to facilitate users, with little knowledge about Linux systems and networks, to be able to set up their own distributed parallel server, with academic or research objectives, and tight budget. Initially, an analysis of the needs of the project was done. Once known, they were selected separately, testing their operation and configurations. Once the established requirements had been met, it was automated using scripts. In order to carry out the tests in a real environment and close to the one where the end user will be, a small-scale cluster was set up in order to carry out the relevant tests. The results obtained have been mostly successful, allowing a correct automation in the assembly of the entire cluster, with minimal user intervention.

Índex

1	INTRODUCCIÓ	5
1.1	DESCRIPCIÓ	5
1.2	OBJECTIUS	5
2	DESCRIPCIÓ GENERAL	6
2.1	ENTORN I TECNOLOGIES SOFTWARE	6
2.1.1	<i>Sistema operatiu i programació</i>	6
2.1.2	<i>Xarxa i comunicacions</i>	7
2.1.3	<i>Sistema paral·lel i distribuït</i>	7
2.2	ENTORN I TECNOLOGIES HARDWARE	9
2.2.1	<i>ODROID C2</i>	9
2.2.2	<i>Switch/Commutador</i>	10
2.2.3	<i>Targeta de xarxa externa</i>	11
2.3	REQUERIMENTS	11
3	DISSENY	13
3.1	ARQUITECTURA DEL SOFTWARE	13
3.1.1	<i>Diagrama de scripts</i>	14
3.1.2	<i>Diagrama de scripts amb funcions</i>	14
3.2	ARQUITECTURA DE LA XARXA	15
3.3	DISSENY DE LA INTERFÍCIE GRÀFICA	15
3.3.1	<i>Esbós</i>	15
3.3.2	<i>Disseny final</i>	16
3.4	DECISIONS DE DISSENY	20
3.4.1	<i>Configuració del software</i>	20
3.4.2	<i>Ubicació network_lib.sh</i>	21
3.4.3	<i>Fitxers de hosts</i>	21
4	IMPLEMENTACIÓ	22
4.1	SCRIPTS MASTER	22
4.1.1	<i>Generals</i>	22
4.1.2	<i>Localització</i>	28
4.1.3	<i>Xarxa</i>	28
4.2	SCRIPTS SLAVES	35
4.2.1	<i>Generals</i>	35
5	AVALUACIÓ	40
5.1	SERVIDOR MASTER	40
5.2	NODE SLAVE	58
5.3	EXECUCIÓ DE PROVA SOBRE EL CLÚSTER DE CODI PARAL·LELITZAT AMB MPI ...	75
6	CONCLUSIONS	76
7	RECURSOS UTILITZATS	77
8	ANNEXES	78
8.1	MANUAL D'INSTAL·LACIÓ	78
8.2	CODI	78
8.2.1	<i>init_master.sh</i>	78
8.2.2	<i>conf_network_master.sh</i>	82
8.2.3	<i>iptables.sh</i>	83
8.2.4	<i>install_slurm.sh</i>	84
8.2.5	<i>set_vnc_password.sh</i>	86
8.2.6	<i>network_lib.sh</i>	86
8.2.7	<i>locale.sh</i>	89

8.2.8	<i>dhcp_script.sh</i>	90
8.2.9	<i>add_slave.sh</i>	92
8.2.10	<i>init_slave.sh</i>	92
8.2.11	<i>check_gui_dependencies.sh</i>	94
8.2.12	<i>gui_dependencies.sh</i>	94
8.2.13	<i>installer_gui.pyw</i>	95

Índex de taules

TAULA 1: RELACIÓ ENTRE SLURM I ALGUNS DELS SEUS COMPLEMENTS.....	8
TAULA 2: VELOCITATS DE LECTURA I ESCRITURA DE LES MEMÒRIES EMMC [3].....	10
TAULA 3: CLASSES DE IP PRIVADA I EL SEUS RESPECTIUS RANGS.....	30
TAULA 4: CONFIGURACIÓ REGLES <i>IPTABLES</i>	35

Índex de figures

FIGURA 1: PCB ODROID C2 [2]	9
FIGURA 2: TOPOLOGIA DE COMMUTADORS EN ESTRELLA.	11
FIGURA 3: DIAGRAMA DELS SCRIPTS QUE FORMEN EL SOFTWARE.	14
FIGURA 4: DIAGRAMA DELS SCRIPTS QUE FORMEN EL SOFTWARE, AMB FUNCIONS.....	14
FIGURA 5: ARQUITECTURA DE LA XARXA ON S'HA DESENVOLUPAT EL SOFTWARE.	15
FIGURA 6: ESBÓS DEL DISSENY DE LA INTERFÍCIE GRÀFICA.....	16
FIGURA 7: PANTALLA DE BENVINGUDA DE LA INTERFÍCIE GRÀFICA.....	16
FIGURA 8: PANTALLA D'INSTAL·LACIÓ DE LA INTERFÍCIE GRÀFICA.....	17
FIGURA 9: PANTALLA D'OPCIONS AVANÇADES DE LA INTERFÍCIE GRÀFICA.....	18
FIGURA 10: PLANIFICADORS DE LA CUA DE PETICIONS SLURM [11].	19
FIGURA 11: PANTALLA DE SLURM DE LA INTERFÍCIE GRÀFICA.....	20
FIGURA 12: MISSATGE D'ERROR PER FALTA DE CLAU AL ACTUALITZAR EL SISTEMA.	23
FIGURA 13: COMPILACIÓ DE CODI DE PROVA AL CLÚSTER.	75
FIGURA 14: EXECUCIÓ DE CODI DE PROVA AMB 2 SUBTASQUES.	75
FIGURA 15: EXECUCIÓ DE CODI DE PROVA AMB 4 SUBTASQUES.	75
FIGURA 16: EXECUCIÓ DE CODI DE PROVA AMB 8 SUBTASQUES.	75

1 Introducció

1.1 Descripció

Amb la millora dels processadors en els últims anys cada cop s'ha fet més important optar per avançar cap a una execució de codi paral·lel, ja que, estem arribant als límits de la llei de Moore en la litografia de fabricació, i els processadors actuals ja funcionen a unes freqüències de rellotge en les que cada cop es més difícil dissipar la calor generada pels transistors. Degut a aquesta sèrie de límits van aparèixer els processadors multinucli (memòria compartida) que consisteixen en múltiples processadors (anomenats nuclis o cores) encapsulats en un únic circuit integrat, permetent executar múltiples instruccions de forma totalment paral·lela, com a inconvenient tenim que presenta poca o nul·la escalabilitat horitzontal, ja que, per ampliar el nostre multiprocessador haurem de reemplaçar-ho per un de nou amb major nombre de nuclis, degut a això tenim els anomenats clústers (memòria distribuïda) que consisteixen en un conjunt de computadors connectats entre si per treballar com si es tractessin d'un únic computador amb un nombre de nuclis prou gran. Per augmentar el nostre clúster i per tant el nombre de nuclis, només serà necessari connectar i configurar noves màquines, és en aquest últim punt on entra aquest treball de fi de grau, en facilitar a l'usuari el muntatge i la configuració de servidors paral·lels distribuïts, ja que requereixen de coneixements avançats en sistemes Linux i xarxes informàtiques, que se l'hi podrien dificultar a un usuari no experimentat que volgués executar software paral·lelitzat, com per exemple en l'àmbit científic o educatiu.

Per tal de facilitar a l'usuari aquesta feina s'ha dissenyat un software basat en *bash* y *python*, aquest últim llenguatge de programació només s'utilitza per la interfície gràfica, on es poden trobar tres apartats, el primer és el més bàsic i l'únic a fer servir per a la majoria d'usuaris, serveix per configurar la zona horària, l'idioma de les màquines i el *layout* del teclat, molt semblant als apartats bàsics en la instal·lació d'un sistema operatiu Linux com per exemple Ubuntu, sent un dels més coneguts. Per altra banda tenim la part de scripts *bash* que és transparent a l'usuari i és l'encarregada de instal·lar el software necessari per al seu funcionament, sent uns dels principals: *dnsmasq*, *slurm work load manager*, *network file sistem* i *ssh*, que detallarem en apartats posteriors.

Una altra part molt important d'aquest treball i que apareix en el nom del mateix és el hardware escollit un SBC¹ (*Single Board Computer*) relativament econòmic basat en ARM², s'ha escollit aquest computador i no un altre per la seva relació qualitat preu, el que el fa molt adient de fer servir en l'àmbit acadèmic.

1.2 Objectius

Els objectius acadèmics d'aquest treball es basen en aprofundir, repassar i millorar els coneixements adquirits durant el grau d'enginyeria informàtica sobre sistemes operatius Linux i xarxes de dades, en conjunt amb les assignatures d'arquitectura de computadors i computació paral·lela i massiva. A més entendre com funcionen els clústers o sistemes distribuïts i com muntar-los. També hi ha una part molt important pel que fa la programació estructurada de scripts *bash* i *python*.

¹ Single Board Computer, és un ordinador complet en un sol circuit.

² Advanced Risc Machine, és una arquitectura RISC de processadors.

2 Descripció general

2.1 Entorn i tecnologies software

2.1.1 Sistema operatiu i programació

2.1.1.1 Ubuntu

Ubuntu és una distribució del sistema operatiu mononucli de codi lliure GNU³/Linux, i a l'hora basada en la distribució Debian. És molt conegut com a sistema d'escriptori per ser amigable amb l'usuari no familiaritzat amb sistemes Linux.

S'ha decidit fer servir aquest i no un altre perquè està ben integrat amb el hardware que es farà servir. Podent trobar una imatge amb Ubuntu Mate a la web del fabricant del hardware.

2.1.1.2 Bash

Bash és un intèrpret de comandes per a Linux, per poder interactuar amb el sistema operatiu i els programes a través de comandes escrites en comptes d'una interfície gràfica. A més es pot utilitzar per automatitzar tasques, com per exemple instal·lar i configurar programes, per aquest motiu s'ha decidit utilitzar-ho en aquest projecte.

2.1.1.3 Python

Python es un llenguatge de programació interpretat, d'alt nivell i de propòsit general, es caracteritza per tindre una llegibilitat molt fàcil, també es multiparadigma perquè amb ell podem programar de forma seqüencial, orientada a objectes o també funcional. En aquest projecte s'ha fet servir per desenvolupar la interfície gràfica, mitjançant la amb la llibreria Tkinter.

El codi s'ha programat per Python 3.6.9, sent més modern i tenint en compte que Python 2.7 és troba sense suport des del 1 de gener de 2020.

2.1.1.4 Tkinter

Tkinter és una llibreria de software orientada a objectes per a Python que ens permet crear interfícies gràfiques de forma ràpida i fàcil mitjançant objectes, exportable als sistemes operatius Windows i Linux.

S'ha decidit fer servir aquesta llibreria i no una altra perquè permet crear la interfície gràfica de forma fàcil per a Linux, a més tindre una GUI⁴ facilita la interacció amb l'usuari poc experimentat en la terminal i les comandes.

³ GNU is Not Unix, és un conjunt d'eines *software*, compatibles amb sistemes *Unix*.

⁴ Graphical User Interface, és un programa informàtic que actua com interfície per a l'usuari utilitzant un conjunt d'imatges i objectes.

2.1.2 Xarxa i comunicacions

2.1.2.1 Dnsmasq

Servidor DNS⁵ i DHCP⁶ de codi lliure, lleuger i fàcil de configurar, per a entorns petits.

S'ha decidit fer servir pel seu baix consum de recursos del sistema, perquè és de codi lliure i perquè es fàcil de configurar. A més amb la seva propietat de poder executar un script cada cop que els clients DHCP interactuen amb el servidor DHCP ha facilitat la preparació dels nous nodes afegits al clúster de forma automàtica.

2.1.2.2 SSH

Protocol i software que serveix per accedir de forma remota a altres computadors de forma segura, ja que les comunicacions estan xifrades, també permet executar comandes de terminal en remot i copiar fitxers i directoris per la xarxa.

Per l'autenticació permet utilitzar tant la contrasenya de l'usuari en el sistema o sistemes de clau asimètrica com RSA⁷.

En el projecte es fa servir per comunicar el *master* amb els *slaves* i així poder instal·lar i configurar el software necessari en aquests.

2.1.2.3 Network File System

Protocol de compartició de fitxers per a xarxes locals, on els clients NFS⁸ poden accedir als directoris i fitxers compartits del servidor NFS com si formessin part del seu propi medi de emmagatzematge, accedint des del seu arbre de directoris.

En aquest projecte s'utilitza per compartir el directori home del node master entre tots els slaves i que d'aquesta forma puguin accedir directament als fitxers a executar en paral·lel. A banda s'utilitza per compartir els fitxers de configuració de Slurm

2.1.3 Sistema paral·lel i distribuït

2.1.3.1 Slurm Workload Manager

Una peça de software clau en un sistema paral·lel distribuït, les seves tres funcions principals son:

- Dona accés al clúster als usuaris de forma excloent o no, durant un cert període de temps, per a que puguin fer-ne us del clúster.
- Proporciona unes eines per a que els usuaris puguin executar i monitorar treballs en un conjunt de nodes del clúster.
- Implementa un sistema de cues de treballs pendents, per quan més d'un usuari vol accedir al clúster i els nodes estan ocupats.

⁵ Domain Name System, sistema amb l'objectiu de traduir adreces IP a noms de domini, entre d'altres.

⁶ Dynamic Host Configuration Protocol, protocol per proporcionar adreces IP al nodes d'una xarxa, de forma dinàmica.

⁷ Rivest, Shamir i Adleman, és un sistema criptogràfic de clau pública.

⁸ Network File System, és un sistema d'arxius distribuïts en una àrea local.

Slurm[6] [5] és un software de codi lliure, pensat per a que executar tasques en un sistema distribuït sigui quasi tan senzill com fer-ho en un computador comú. Entre d'altres es caracteritza per ser petit i simple, altament escalable, amigable amb l'administrador del sistema, portable i tolerant a fallades mes conegut com *failure tolerance*.

A més les seves funcions són ampliables mitjançant una gran varietat de complements, com poden ser:

Slurm Kernel					
Plugin:	Autenticació	MPI	Punt de control i restauració	Topologia	Emmagatzematge comptabilitat
Nom:	Munge	mvapich	BLCR ⁹	Tree	MySQL

Taula 1: Relació entre Slurm i alguns dels seus complements.

Slurm consta de dues parts molt importants a l'hora de comunicar els nodes del clúster, tenint dos servei o dimonis anomenats slurmd i slurmctld, el primer s'instal·la al *controller* que en aquest projecte s'ha anomenat master i és l'encarregat de dirigir el clúster, repartint la feina entre els nodes, el segon s'instal·la als nodes de computació i la seva única feina es esperar a que el *controller* l'hi doni feina per executar. Aquest model de comunicació s'anomena *master/slave*[9], encara que al darrer segle degut al seu nom controvertit també és conegut com *primary/secondary*, etc.

Pot haver més d'un *controller* a un mateix clúster, permetent que si el primer cau, un segon el pugui reemplaçar (*Failure tolerance*).

2.1.3.2 MPICH

MPICH és una implementació de MPI (Message Passing Interface), es caracteritza per la seva alta portabilitat admetent diferents arquitectures de processadors i sistemes de comunicació en la xarxa, des de ús comú com Ethernet fins a d'altres dissenyats expressament per treballar en clústers com Myrinet, i proporcionar un alt rendiment.

En primera instància s'havia pensat en treballar amb OpenMPI, però degut ha problemes de compatibilitat amb el processador ARM-v8 s'ha optat per MPICH.

2.1.3.3 Munge

Munge[12] és un software que funciona com a servei d'autenticació sense necessitat de privilegis de root, dissenyat específicament per ser utilitzat en HPC¹⁰ i per tant altament escalable degut al seu entorn. És necessari crear una clau simètrica i compartir-la amb tots

⁹ Berkeley Lab Checkpoint/Restart, proporciona punts de control i reinici en sistemes Linux.

¹⁰ High-performance Computing, és l'aprofitament de tecnologies com la computació paral·lela i distribuïda, en l'objectiu de resoldre problemes computacionalment avançats.

els serveis anomenats *munged*, allotjats als nodes de la xarxa, també és un requisit previ que tots els computadors tinguin el mateix GID¹¹ i UID¹² per poder validar-se.

Slurm utilitza Munge com a plugin per autenticar els nodes de la xarxa entre sí, identificant l'usuari que origina un missatge enviat.

2.2 Entorn i tecnologies hardware

2.2.1 ODROID C2



Figura 1: PCB ODROID C2 [2]

La part central del Hardware d'aquest projecte son els computadors, en aquest cas anomenats SBC perquè tot el hardware necessari està integrat a la placa base, a excepció del medi físic d'emmagatzematge.

Especificacions del SBC utilitzat:

- CPU¹³: ARM Cortex-A53 (ARM v8) de 4 nuclis a 1,5 GHz de 64 bits.
- GPU¹⁴: Mali-450 [4] compatible amb OpenGL ES 1.1/2.0 i OpenVG 1.1.
- RAM¹⁵: 2GB DDR3 SDRAM.
- NIC¹⁶: Gigabit Ethernet (1000 Mb per segon).
- USB: 4 x USB 2.0
- HDMI: HDMI 2.0

¹¹ Group ID, és l'identificador d'un grup en sistemes UNIX.

¹² User ID, és l'identificador de l'usuari en sistemes UNIX.

¹³ Central Processing Unit, és la part central del computador, encarregada d'interpretar les instruccions.

¹⁴ Graphics Processing Unit, és un coprocessador dedicat al processament gràfic.

¹⁵ Random Access Memory, és la memòria de treball a un computador

¹⁶ Network Interface Card, és el component *Hardware* encarregat de connectar un ordinador a una xarxa.

ODORID C2 proporciona dues possibilitats d'emmagatzematge:

- Micro-SD UHS-1: Format d'emmagatzematge estàndard de tipus memòria flash, recolzat per l'associació de targetes SD (Panasonic, SanDisk Corporation i Toshiba Corporation)[1] . Amb una velocitat mínima de 10 MB/s surt perdent contra eMMC5.0, encara que té un preu de mercat molt més reduït i es fàcil de trobar.
- EMMC5.0: Format d'emmagatzematge estàndard amb una connexió propietària de Hardkernel, fabricants del SBC, també és una memòria flash, encara que molt més ràpida que Micro-SD UHS-I.

En la següent taula podem observar les velocitats obtingudes en els test de rendiment proporcionats pel fabricant, tots els resultats són en MB/s, sent molt més ràpides que una memòria Micro-SD:

		Samsung	Toshiba	Sandisk
8G	Write	45.4	21.9	N/A
8G	Read	113	148	N/A
16G	Write	80.1	N/A	25.6
16G	Read	126	N/A	153
32G	Write	124	N/A	98.7
32G	Read	125	N/A	153
64G	Write	124	83.7	107
64G	Read	124	153	153

Taula 2: Velocitats de lectura i escriptura de les memòries EMMC [3]

S'ha escollit aquest computador i no un altre, pel seu reduït preu, mida i consum elèctric, sacrificant la seva potència de càlcul en comparació amb processadors x86-64 amb arquitectura CISC. També es important destacar que ODROID-C2 no necessita de dissipació activa com ventiladors.

Una altra opció seria raspberry pi 3, però s'ha tingut en compte que ODROID-C2 té un millor hardware i que raspberry no té suport per memòries EMMC, ni gigabit Ethernet fins la seva quarta generació.

2.2.2 Switch/Commutador

Dispositiu físic de propòsit específic és l'encarregat d'interconnectar els nodes de la xarxa, unint-la i formant-la[15] . El commutador s'encarrega de comunicar tots els nodes de la xarxa enviant els paquets al seu destinatari, a diferència del hub guarda dinàmicament a quina dels seus ports es troba cada MAC¹⁷ de la xarxa, quan encara no les coneix envia el paquet a tots els ports i només respon el propietari. Amb aquest aprenentatge i coneixement de la xarxa evita congestions i problemes de seguretat.

¹⁷ Media Access Controller, és un identificador únic per a cada NIC.

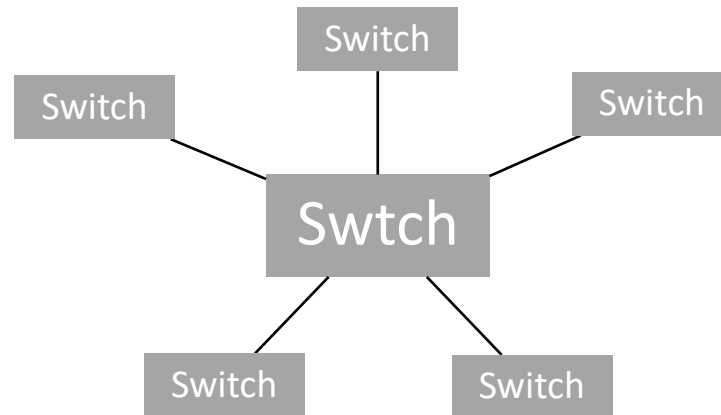


Figura 2: Topologia de commutadors en estrella.

En clústers molt petits només serà necessari un únic commutador, però a mesura que creixi el número de nodes, s'hauran de interconnectar els commutadors entre si, podent-se fer en diverses topologies segons les necessitats, la utilitzada en el nostre cas serà la topologia en estrella, on trobem un commutador central que uneix i comunica tots els commutadors.

També es pot trobar solucions de commutadors professionals on podem connectar una gran quantitat de nodes.

Pel que fa a les connexions entre commutadors i els computadors que formen la xarxa seran necessaris cables que permetin aprofitar la tecnologia Gigabit Ethernet[14] de la interfície de xarxa que porta ODROID C2, per tant com a mínim seria necessari fer servir cables de categoria Ethernet Cat 5e amb velocitat de 1.000Mbps i freqüència de 100MHz, o categories superiors. També seran necessaris commutadors amb tecnologia Gigabit Ethernet per poder treure la màxima velocitat en les comunicacions del sistema paral·lel i distribuït, sent un factor molt important i limitant el rendiment de les execucions de codi paral·lel.

2.2.3 Targeta de xarxa externa

Degut a que ODROID C2 només compta amb una targeta de xarxa, és necessari afegir al conjunt de hardware necessari per muntar el clúster una targeta de xarxa externa amb interfície USB, seria preferible que fos Gigabit Ethernet amb USB 3.0, però degut a que els ports USB amb els que compta són 2.0 no s'aprofitaria tota la velocitat que pot donar Gigabit Ethernet.

2.3 Requeriments

- Comunicar tots els nodes de la xarxa.
 - Donar servei DHCP a la xarxa interna.
 - Donar servei DNS a la xarxa interna.
 - Generar i compartir clau pública entre tots els nodes de la xarxa.
- Instal·lar software per la comunicació per pas de missatge.
- Instal·lar software de sistema de cues per a que múltiples usuaris puguin accedir al clúster.
- Afegir una capa de seguretat a les xarxes mitjançant un Firewall.
- Donar accés al directori *home* del *master* a tots els nodes de la xarxa.

- Permetre als usuaris modificar la localització del sistema operatiu del clúster (zona horària, idioma i teclat).
- Permetre als usuaris escollir la classe i rang de la IP¹⁸ privada.
- Permetre als usuaris configurar el software *Slurm* segons les seves necessitats.
- Permetre als usuaris accedir mitjançant interfície gràfica al node *master* del clúster.

¹⁸ *Internet Protocol, és un valor numèric que identifica una interfície de xarxa.*

3 Disseny

3.1 Arquitectura del software

El disseny o arquitectura del software es divideix en dos parts, la primera és la instal·lació del servidor anomenat master i s'inicia amb *init_master.sh* o *installer_gui.pyw* si es fa servir la interfície gràfica, i només s'executarà un cop per instal·lar i configurar tot l'entorn del servidor. Després de *init_master.sh* aquesta primera part està formada per *network_lib.sh*, fent-se servir com llibreria en 4 dels scripts per operacions amb la IP i la mascara de xarxa, com per exemple obtindrà la ip de la xarxa a partir de la ip i mascara donades per l'usuari, després tenim, *conf_network_master.sh*, la seva funció és configurar tots els fitxers del master relacionats amb la xarxa, com activar l'encaminament de paquets, *forwarding* en anglès. Per altra banda tenim *iptables.sh* on la seva única funció és afegir les polítiques i regles del tallafocs del servidor, per últim tenim els scripts *install_slurm.sh* que instal·la i configura slurm, *set_vnc_password.sh* que afegeix una contrasenya al servidor VNC¹⁹ i *locale.sh* que funciona com una llibreria amb l'únic objectiu de modificar els paràmetres de localització software del *master*.

La segona part es la instal·lació dels demés nodes de la xarxa, anomenats *slaves*, aquesta part dissenyat per a que sigui automàtica, sense intervenció per part de l'usuari, això s'ha aconseguit mitjançant un mecanisme del servidor DHCP inclòs a *dnsmasq*, amb el que un cop rep una sol·licitud d'assignació de IP per part d'un nou client de la xarxa, aquest executa un script per guardar el client donant-li una IP fixa, incloent un nom de host a la xarxa, a més es comunica mitjançant SSH²⁰ amb el client, hi executa un script en remot per tal de iniciar-lo, instal·lant tot el software necessari i configurant-lo, els scripts que intervenen en aquesta segona part són el *dhcp_script.sh*, sent el script que s'ha explicat abans que executa *dnsmasq*, *add_slave.sh* aquest es l'encarregat de comunicar-se amb el client i executar l'últim script, anomenat *init_slave.sh* que té la funció de preparar el client per a que sigui un nou *slave* dintre del clúster i poder atendre les peticions d'execució de codi, que venen del *master*, *locale.sh* és el mateix script que fa servir el *master*, però pel *slave*.

¹⁹ Virtual Network Computing, és un programa que permet veure les accions d'un ordinador remot.

²⁰ Secure Shell, és un programa o protocol que permet l'accés remot a un altre ordinador.

3.1.1 Diagrama de scripts

Al següent diagrama podem veure com interactuen entre si els scripts del projecte, amb un total de 13, també podem observar el llenguatge que utilitzen, sent bash scripting per a tots, a excepció de *installer_gui.pyw* que està escrit en python 3. Les fletxes indiquen que el script A crida al script B. El *dhcp_script.sh* és cridat pel servidor DHCP de *dnsmasq*.

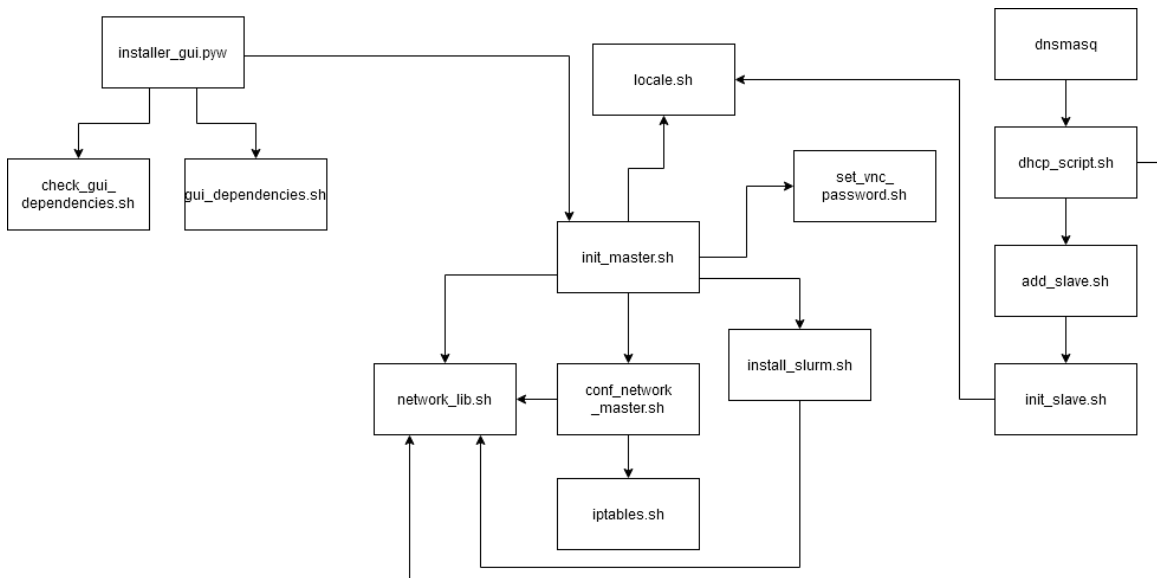


Figura 3: Diagrama dels scripts que formen el software.

3.1.2 Diagrama de scripts amb funcions

Aquest diagrama mostra la mateixa informació que l'anterior afegint les funcions de cada script, entenent funció no només com a mètodes i funcions de codi si no com a tasques que formen part del script, encara que no estiguin encapsulades.

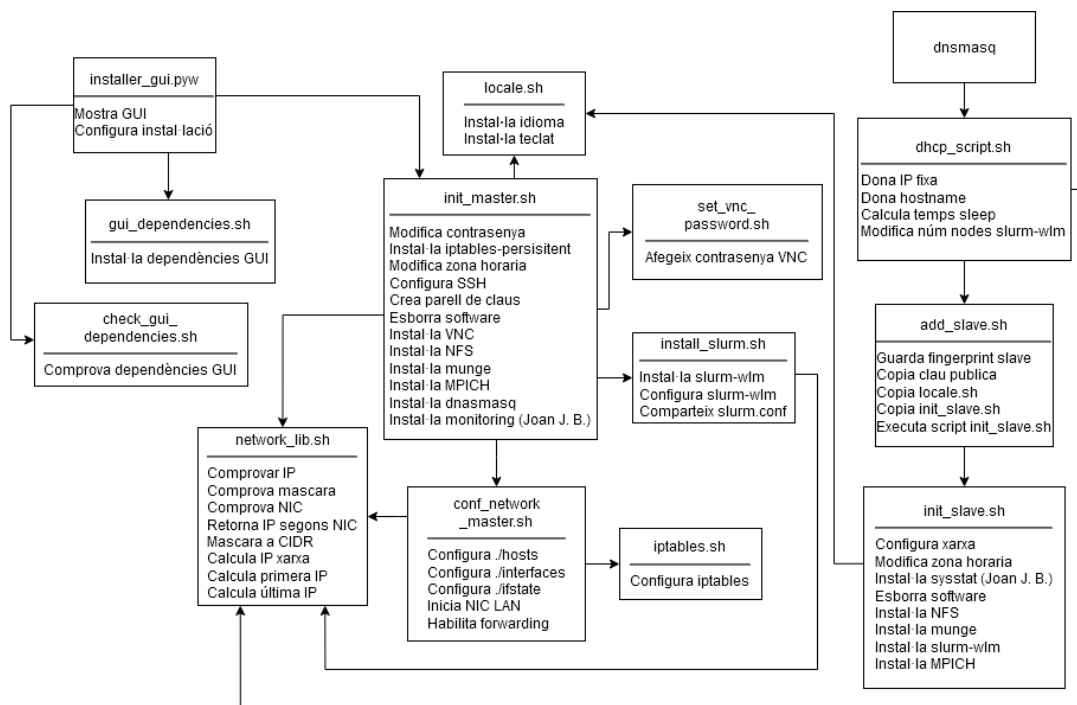


Figura 4: Diagrama dels scripts que formen el software, amb funcions.

3.2 Arquitectura de la xarxa

La xarxa està formada per un node que fa de servidor paral·lel i distribuït, aquest també fa les funcions de Firewall per protegir tot el conjunt d'atacs externs, per altra banda hi ha un o més commutadors segons el numero de nodes que s'hagin de connectar i el numero de ports que tingui cada un dels commutadors, d'altra banda hi ha els nodes de computació anomenats *slaves*. El *master* fa de *router* entre la xarxa interna, on es troben els *slaves* i internet.

El Firewall configurat al node *master* s'anomena iptables i en aquest cas és de tipus *packet-filtering statefull*, per tant actua a les capes 3 (Xarxa) i 4 (Transport) del model *TCP/IP*, aplica un conjunt de polítiques i regles a cada paquet que entra i surt de la xarxa i el reenvia/*forwarding*. *Statefull* vol dir que per cada paquet entrant o sortint, el *firewall* guarda el seu estat, d'aquesta manera pot ser que no tinguem una regla que permeti la sortida de la xarxa per un port, però al haver hagut una connexió acceptada d'entrada en aquell mateix port, pot ser que si que els paquets surtin per aquell port, per continuar amb la comunicació establerta.

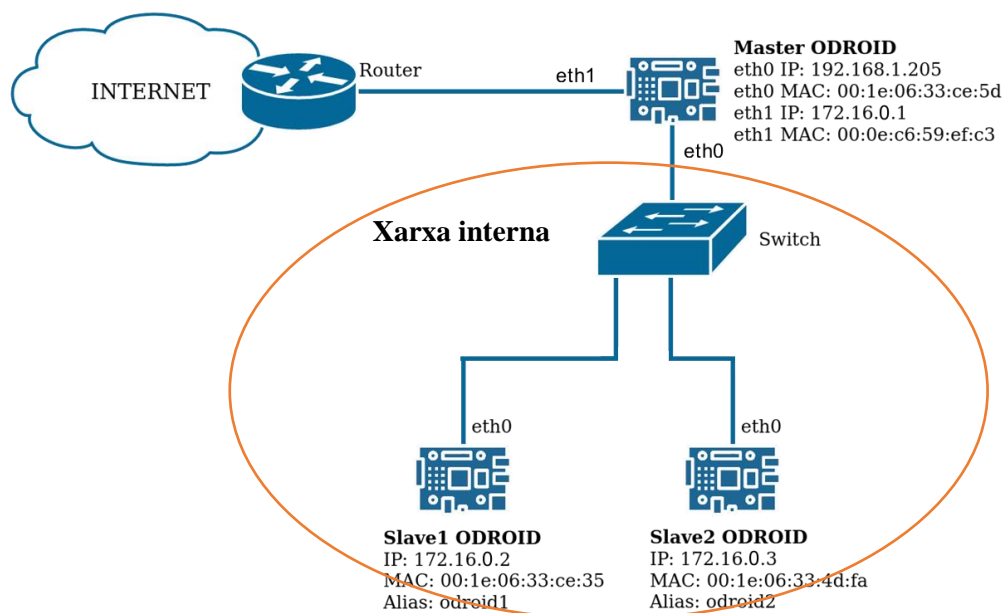


Figura 5: Arquitectura de la xarxa on s'ha desenvolupat el software.

3.3 Disseny de la interfície gràfica

3.3.1 Esbós

Abans de començar la interfície gràfica es va pensar en fer un esbós amb el software gràfic Inkscape, pensant en com s'estructuraria la interfície gràfica i quins apartats tindria.

A la versió final es va reordenar completament i esborrar algun del contingut de la pestanya de Install, esborrant la opció de "Security level" i la de "Max number of nodes", fent que el nombre màxim de nodes no estigui limitat i es vagi incrementant dinàmicament des de 1 fins a N, només estaria limitat segons el rang de IP escollit. El menú de d'alt i els seus apartats si que es van conservar.

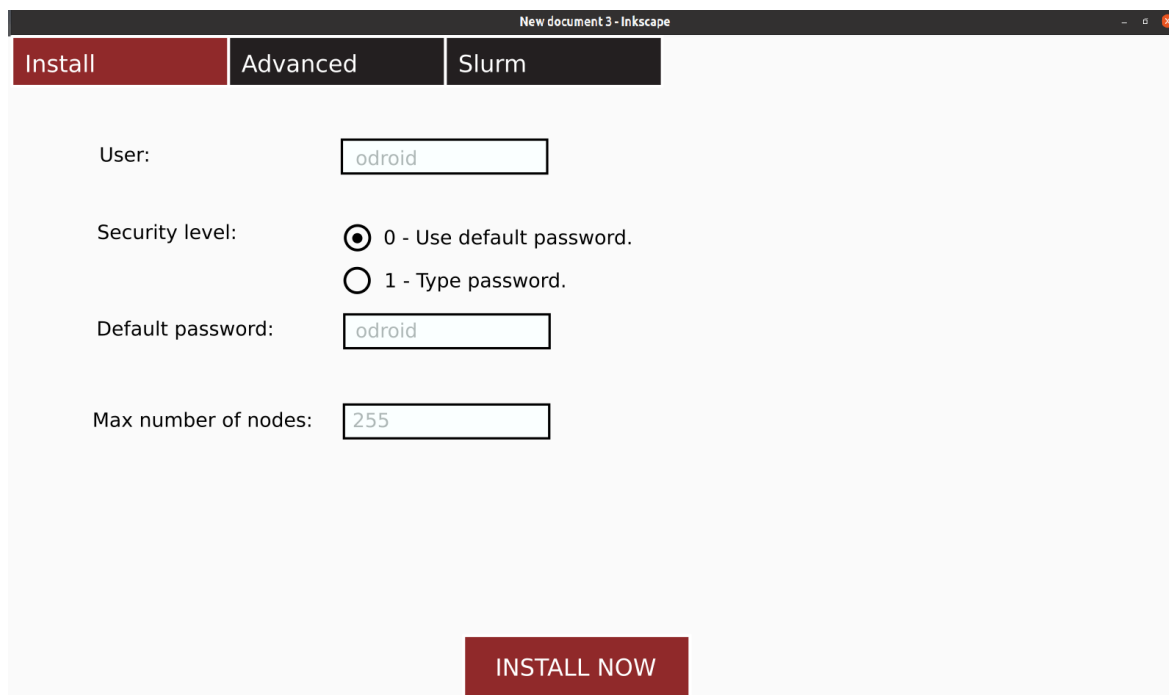


Figura 6: Esbós del disseny de la interfície gràfica.

3.3.2 Disseny final

3.3.2.1 Pantalla de benvinguda

Com a detall, es va decidir afegir una pantalla de benvinguda al inici del programa, amb un nom i logotip inventats, també es mostrant el logotip de la URV a la part inferior dreta, aquesta pantalla no és una simple imatge mostrada, si no que està muntada i es mostra mitjançant tkinter.

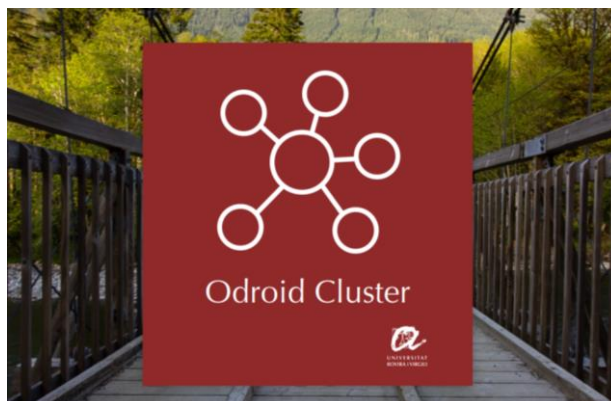


Figura 7: Pantalla de benvinguda de la interfície gràfica.

3.3.2.2 Pantalla d'instal·lació

Al disseny final es va decidir esborrar o moure tot el contingut que s'havia pensat durant l'esbós i es van afegir uns apartats de localització, semblant als que es poden veure a la instal·lació de un sistema operatiu com per exemple Ubuntu. Permetent escollir la zona horària i l'idioma mitjançant dos menús desplegable i podent escollir el *layout* del teclat, segons l'idioma, i la seva variant, per exemple, Spanish – Catalan.

Tota la informació de zones horàries, idiomes i teclats, s'extreu de fitxers del sistema o comandes específiques, en concret:

- Zona horària: `timedatectl list-timezones`
- Idioma: `/usr/share/i18n/SUPPORTED`
- Teclat: Diversos fitxers emmagatzemats al directori `/usr/share/X11/xkb/symbols/`

La modificació de teclat només serveix per la interfície gràfica, ja que en connexions SSH estàndard com les que he fet servir durant el desenvolupament del treball, utilitzes el teclat definit al teu sistema operatiu.

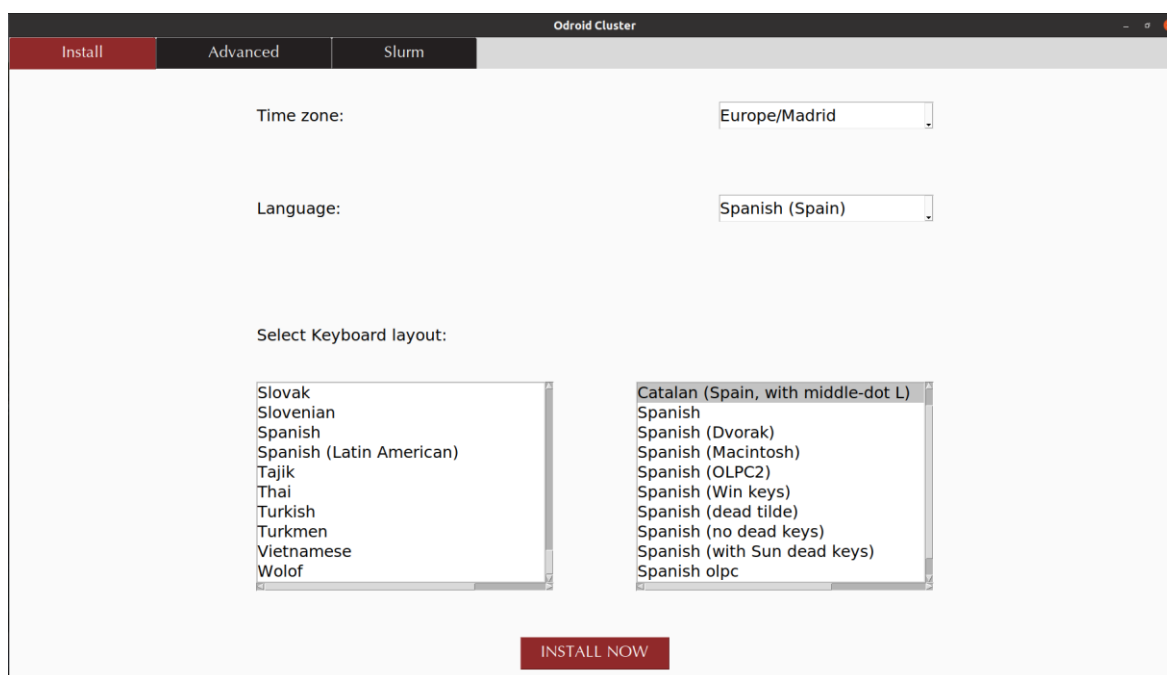


Figura 8: Pantalla d'instal·lació de la interfície gràfica.

3.3.2.3 Pantalla d'opcions avançades

En aquest apartat s'han inclòs diverses opcions que s'aniran comentant de dalt a baix.

Pels primers tres apartats tenim tres entrades de text en les que podem modificar l'usuari, `hostname` i contrasenya per defecte que esperen els scripts de instal·lació, s'ha pensat en aquest apartat per si més endavant els modifiquen a les imatges de Ubuntu Mate que proporciona el fabricant, o per adaptar el software a altres plataformes.

Els dos següents apartats permeten modificar els DNS que consultarà el servidor DNS local, en aquest cas `dnsmasq`, si se l'hi fan peticions de noms de domini externs als locals, com per exemple `urv.cat`.

Durant la instal·lació es copien uns scripts que mes tard es fan servir per accedir, instal·lar i configurar els `slaves` que es van connectant, per aquest motiu ha d'existir un directori on sempre es pugui accedir a aquests scripts, s'ha permès modificar el nom i `path` d'aquest directori, deixant per defecte `/opt/scripts`

Si continuem, al setè apartat trobarem un element gràfic de tipus `checkbox` juntament amb una entrada de text, si l'usuari marca el `checkbox` el `master` i els `slaves` instal·laran les

seves últimes versions dels paquets de software que trobin als repositoris de software afegits per defecte. S'ha decidit incloure una entrada de text, només numèrica per afegir un temps d'espera entre cada instal·lació i no saturar la xarxa, ja que el *master* fa coll d'ampolla en la xarxa interna a l'hora de sortir a internet.

Per últim, es va afegir, l'opció d'escollir la classe de IP privada que configurarà la xarxa interna i quin rang es farà servir, mitjançant una IP que escull l'usuari limitant-se a la classe que ha escollit, i una mascara de xarxa en notació CIDR²¹, arribant fins a 31, ja que 32 no permetria cap host a la xarxa.

En tots aquests apartats s'ha intentat evitar i controlar l'error humà, limitant-lo a opcions correctes.

The screenshot shows the 'Advanced' configuration tab for 'Odroid Cluster'. It contains the following fields and options:

- Default odroid user:
- Default odroid hostname:
- Default odroid password:
- Upstream DNS server 1:
- Upstream DNS server 2:
- Scripts directory:
- Upgrade cluster
- Upgrade time:
- Private IP: Class A Class B Class C
- IP: . . .
- Mask:

Figura 9: Pantalla d'opcions avançades de la interfície gràfica.

3.3.2.4 Pantalla de *Slurm*

Per finalitzar el disseny de la interfície gràfica, s'ha fet una pestanya al menú dedicada a poder configurar algunes de les opcions del servidor paral·lel i distribuït.

A la Figura 11 es pot trobar una entrada de text, que permet introduir el temps màxim d'execució d'un programa dins del clúster. Donant la possibilitat de introduir un valor enter que es considerarà en minuts, o la paraula INFINITE per indicar que les execucions no tenen límit de temps. En cas de que s'introdueixi un valor numèric amb decimals, es considerarà la seva part entera.

²¹ Classless Inter-Domain Routing, és una representació compacta d'una adreça IP i els seus sufixos.

A l'apartat de mes a baix es troba un grup de radio buttons, que ens permeten seleccionar el tipus de planificador, aquest actua a l'hora de que un o més usuaris intentin fer múltiples execucions sobre el clúster.

Planificador FIFO²²: Aquest planificador limita les execucions simultànies dins del clúster en només una, si un usuari es troba executant un codi i si abans de que acabi la execució ell mateix o un altre usuari intenta executar el seu software dins del clúster, el planificador *encuarà* la seva petició, a la cua de Slurm i aquesta esperarà a que termini l'execució actual, si van arribant múltiples peticions d'execució, s'aniran *encuant*, a la espera de poder ser ateses.

Planificador FIFO amb farciment/backfill: Aquest planificador actua igual que l'anterior, amb la diferència de que si un usuari intenta executar software al clúster, mentre hi ha una altra tasca en procés d'execució, el planificador consultarà si hi han nodes disponibles, és a dir en estat ociós/idle, en cas de que el numero de nodes disponibles sigui igual al numero de nodes necessari per executar la tasca, aquesta s'executarà juntament amb la que es trobava en execució, és a dir, les tasques amb menys requeriments podran avançar a altres tasques en la cua amb un alt requisit de recursos del cluster, si això no es controla en la configuració del clúster pot arribar a deixar en estat d'espera permanent a les tasques amb un nivell de subtasques major.

Resumint en una sola imatge:

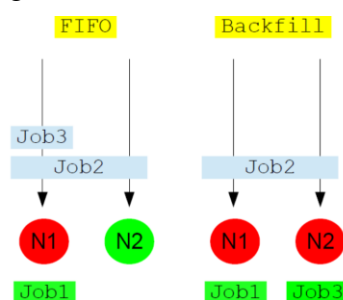


Figura 10: Planificadors de la cua de peticions Slurm [11] .

Per finalitzar l'apartat de Slurm s'ha introduït l'opció d'escollir el mecanisme que farà servir slurmd pel seguiment del processos fills que es van creant durant les execucions als nodes, el més recomanable és utilitzar cgroups i segons afirmen al manual de Slurm, les demés opcions poden fer que slurmd perdi el control dels processos fills.

²² First In First Out. Tipus de planificador on els processos surten per ordre d'entrada.

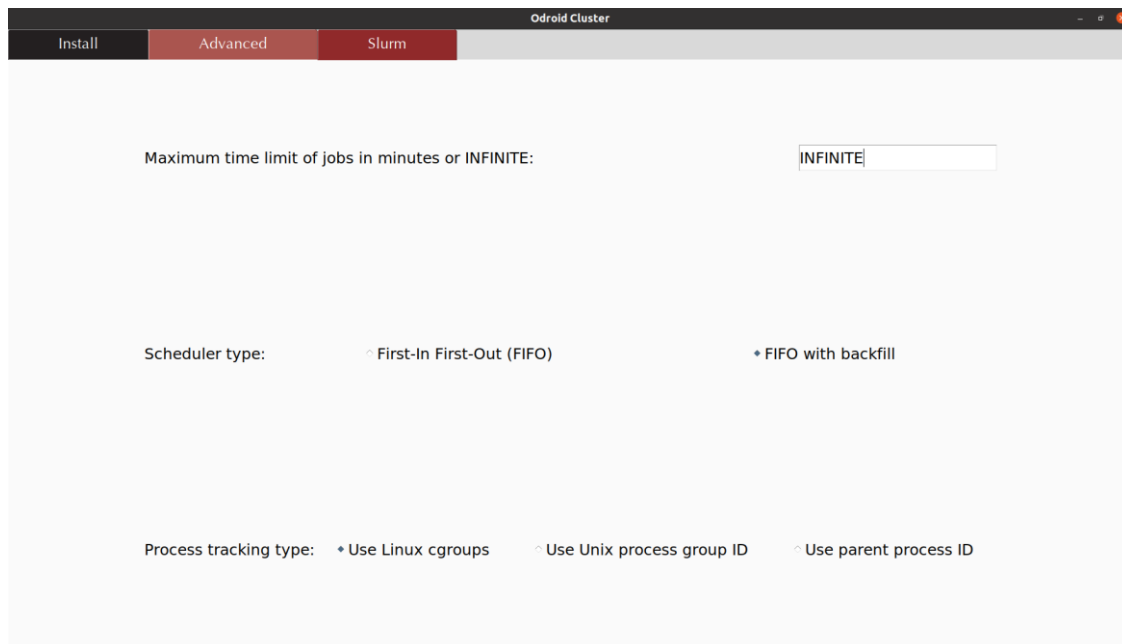


Figura 11: Pantalla de Slurm de la interfície gràfica.

3.4 Decisions de disseny

3.4.1 Configuració del software

Amb l'objectiu d'intentar evitar el *hard-coding* i de poder fer modificacions o retocs al software, amb l'objectiu de permetre adaptar alguns paràmetres a les necessitats de l'usuari, s'ha afegit un fitxer de configuració on accedeixen la gran majoria dels scripts que s'executen al node *master* durant la instal·lació. Pel que fa els nodes *slave* accedeixen a les configuracions necessàries mitjançant arguments que es passen durant la crida per SSH de `init_slave.sh`.

El format del fitxer `odroid_cluster.conf` consta de tres parts, la primera, una paraula clau escrita en majúscules serveix per poder trobar les diferents opcions dins del fitxer, com si es tractes del nom d'una variable, a continuació tenim un separador en forma de signe = i per últim tenim el valor donat a la opció, com si fos el valor que guarda una variable.

NOM=VALOR

A aquest fitxer se l'hi ha donat el nom de `odroid_cluster.conf` i s'ha decidit ubicar al directori `/etc` on es troben la majoria de fitxers de configuració als sistemes Linux.

A continuació es mostra un exemple del possible contingut de `odroid_cluster.conf`:

```
DEFAULT_USER=odroid
DEFAULT_PASSWORD=odroid
HOSTS_NAME=odroid
EXTERNALDNS1=8.8.8.8
EXTERNALDNS2=8.8.4.4
SCRIPTS_DIR=/opt/scripts
UPGRADE=0
```

```

MAX_TIME=INFINITE
SCHEDULER=sched/backfill
TRACKING=proctrack/cgroup
UPGRADE_SLEEP=15
SYS_LANGUAGE=es_ES
SYS_TIMEZONE=Europe/Madrid
IP=172.16.0.1
MASK=255:255:255:0
IP_CLASS=B
LAYOUT=es
VARIANT=cat

```

3.4.2 Ubicació *network_lib.sh*

Degut a que el script *network_lib.sh* es fa servir a múltiples scripts del projecte, s'ha decidit fer-lo servir com una llibreria, és a dir, no s'executa directament amb una crida normal, si no que, es carreguen les seves funcions al script que les necessita, mitjançant la comanda *source*, amb aquest mecanisme el script que ho necessiti pot fer servir les funcions de *network_lib.sh*, com si es tractessin de funcions seves.

Per a que els scripts que fan us de *network_lib.sh* puguin accedir a ell fàcilment, i no hagin problemes per trobar-lo mitjançant la variable *PATH*, s'ha decidit incloure *network_lib.sh* al directori */usr/local/sbin/*. S'ha pres aquesta decisió degut a que el script *dhcp_script.sh* va tindre múltiples problemes per trobar les rutes afegides a la variable de entorn *PATH*.

Per fer la elecció del directori */usr/local/sbin/* i no un altre, s'han tingut en compte els següents criteris:

- És un script que executa l'usuari *root*
- No forma part dels scripts necessaris per engegar el sistema
- No forma part dels directoris que manega el sistema de paquets de *Ubuntu*.

3.4.3 Fitxers de *hosts*

S'ha decidit crear el fitxer */etc/hosts.d/tmp_hosts* i */etc/hosts.d/lan_hosts*, en comptes de */etc/hosts* perquè s'ha deshabilitat l'opció de que *dnsmasq* comparteixi els *hosts* d'aquest últim, per a que no hagin problemes amb els *hosts* que no es volen compartir a la xarxa, i que guarda */etc/hosts*. Per exemple, el servidor assigna a */etc/hosts* el nom *master* a l'adreça de *loopback* (127.0.0.1), però en canvi pels nodes *slaves*, *master* no ha de ser la seva adreça de *loopback*, si no, la IP del servidor dins la xarxa interna.

Les funcions d'aquest fitxers s'expliquen més detalladament a la secció 4 Implementació, però resumidament el fitxer */etc/hosts.d/lan_hosts* guarda la IP del servidor amb el nom de host que comparteix a la xarxa interna (*master*) i el fitxer, */etc/hosts.d/tmp_hosts* guarda les IPs i noms de host assignat a cada *slave*, només de forma temporal.

4 Implementació

4.1 Scripts *master*

4.1.1 Generals

4.1.1.1 *init_master.sh*

El script *init_master.sh* és el principal del projecte, sent de tots l'únic que executa l'usuari, o bé ell mateix mitjançant la terminal o amb la interfície gràfica com intermediària.

Aquest script no té un objectiu específic si no que, fa una gran varietat de tasques instal·lant una gran varietat de software del que ja s'ha parlat al projecte i que serà detallat a continuació, i desinstal·lant d'altres, a més d'aplicar configuracions i cridant a molts dels scripts que apareixen al projecte com es pot observar a la Figura 3 de la pàgina 14.

S'ha tingut en compte que només l'usuari *root* pugui executar el software, ja que quasi totes les accions que s'han de fer necessiten permisos de usuari *root*.

Per començar es deshabiliten les actualitzacions automàtiques del sistema, desinstal·lant el servei *unattended-upgrades* mitjançant la comanda *systemctl*, s'ha pres aquesta decisió degut a que pot sorgir la situació en la que s'executi i deixi bloquejat *dpkg*, això fa que l'execució no pugui funcionar de la forma desitjada, deixant sense instal·lar una part o tots els paquets.

Més tard es creen els fitxers i directoris necessaris per poder dur a terme la instal·lació, per començar crea el directori indicat al fitxer de configuració amb la paraula clau *SCRIPTS_DIR* i copia els scripts *dhcp_script.sh*, *init_slave.sh*, *add_slave.sh*, *locale.sh* i *iptables.sh*, tots aquests a excepció de *iptables.sh* es copien perquè seran necessaris per instal·lar i configurar els nous nodes. També es crea un directori de *logs* a */var/log/odroid_cluster* que faran servir els scripts *add_slave.sh* i *init_slave.sh* per afegir els seus outputs del *stdout* i *stderr*, d'aquesta manera l'administrador del sistema podrà consultar que està passant si afegeix un nou node i no es configura correctament.

Els noms dels arxius que s'afegeixen al directori */var/log/odroid_cluster* són del tipus:

- */var/log/odroid_cluster/add_slave_odroid1.out*
- */var/log/odroid_cluster/init_slave_odroid1.out*

On la primera part del nom, indiquen a quin script pertanyen, o bé *add_slave.sh* o *init_slave.sh* i la següent indiquen a quin node, sent *odroid1* el primer node de tipus *slave* que s'afegeixi i incrementant-se en 1 per cada *slave* nou.

Degut a un error que porta la imatge de Ubuntu de ODROID C2, s'ha d'afegir una clau del servidor de claus de Ubuntu mitjançant la comanda:

```
apt-key adv -v --keyserver keyserver.ubuntu.com --recv-keys
5360FB9DAB19BAC9
```

El codi de la clau faltant s'ha aconseguit a partir del missatge d'error, després de la paraula *EXPKEYSIG*.


```

W: An error occurred during the signature verification. The repository is not updated and the previous
index files will be used. GPG error: http://deb.odroid.in/c2 bionic InRelease: The following
signatures were invalid: EXPKEYSIG 5360FB9DAB19BAC9 Mauro Ribeiro (mdrjr) <
mauro.ribeiro@hardkernel.com>

W: Failed to fetch http://deb.odroid.in/c2/dists/bionic/InRelease The following signatures were
invalid: EXPKEYSIG 5360FB9DAB19BAC9 Mauro Ribeiro (mdrjr) <mauro.ribeiro@hardkernel.com>

W: Some index files failed to download. They have been ignored, or old ones used instead.

```

Figura 12: Missatge d'error per falta de clau al actualitzar el sistema.

Després és modifica el *hostname* (nom de la màquina) del node *master*, passant de *odroid* a *master*, així és més fàcil identificar-lo dins la xarxa i és requisit de slurm indicar un *hostname* diferent per a cada node. Per poder fer-ho s'ha utilitzat la comanda:

```
hostnamectl set-hostname master
```

Una de les primeres coses que fa aquest script és demanar dos cops a l'usuari que introdueixi una contrasenya, per modificar la que porta la imatge per defecte:

```

# Demana la contrasenya dos cops, fins que coincideixin
while [ $pass != $pass2 ]; do
    echo "Enter the new password for the master node:"
    stty -echo
    read -r pass
    stty echo
    echo "Re-enter the new password:"
    stty -echo
    read -r pass2
    stty echo
done

# Modifica la contrasenya de l'usuari root i el per defecte
echo -e "${pass}\n${pass}" | passwd &> /dev/null
echo -e "${pass}\n${pass}" | passwd $master_name &> /dev/null
# Modifiquem el password de vnc
./set_vnc_password.sh "$pass"

```

Com es pot veure a dalt s'ha ubicat en un bucle per evitar que l'usuari fiqui la contrasenya de forma diferent de com l'havia pensat i que després no pugui entrar, a més quan ha d'introduir la contrasenya no es mostra el *echo* per evitar que algú la pugui veure mentre s'escriu. Aquesta contrasenya s'afegeix al usuari *odroid* i al *root*, a més mitjançant un altre script anomenat *set_vnc_password.sh*^[16] s'afegeix com a contrasenya del servidor VNC. La modificació de la contrasenya de VNC és fa en un altre script degut a que, aquest canvi necessita de la interacció amb l'usuari i per evitar-la s'ha utilitzat una comanda anomenada *expect*, d'aquesta manera es pot seguir un diàleg, ficant el que esperes que et demani i el que li respondràs.

A continuació modificarà la zona horària que utilitza el sistema operatiu, amb la comanda `timedatectl set-timezone` i instal·larà el servei *chrony*, encarregat de tindre l'hora actualitzada, mitjançant el protocol NTP²³.

Per mantenir les regles de *iptables* un cop reiniciat el sistema es necessari un servei extra que les carregui al inici, per això s'ha instal·lat *iptables-persistent*, el problema ve quan es vol fer la instal·lació dins d'un script, ja que mostra una interfície gràfica sobre la terminal, anomenada *debconf* amb la que interactuar, per evitar aquest obstacle s'ha hagut d'introduir les següents comandes:

```
echo iptables-persistent iptables-persistent/autosave_v4 boolean true
| debconf-set-selections

echo iptables-persistent iptables-persistent/autosave_v6 boolean true
| debconf-set-selections
```

El que fan és preconfigurar les respostes afegint-les mitjançant la comanda *debconf-set-selections*[19] , d'aquesta manera *debconf* ja no apareixerà per preguntar a l'usuari.

Després es crida el script de xarxa que s'explicarà en detall en apartats posteriors, aquest retorna mitjançant un *echo* alguns valors necessaris a posteriori, com la IP, la mascara de xarxa, la interfície de xarxa que connecta amb internet i la interfície de xarxa que comunica amb la xarxa interna. Aquests valors arriben dins d'un *string* separats per ; per separar-los es podrien haver fet servir múltiples *cut*, però per fer-ho més senzill i treballar amb una *array*, s'ha utilitzat la comanda *IFS=';' read -a net_array <<< "\$result"[17] [18]* , per explicar-la és pot dividir en tres parts, la primera, on podem veure *IFS=';'* indica que el separador que farà servir internament la terminal serà el punt i coma, la segona és la comanda *read*, que és pot fer servir com un *scanf* de C però en aquest cas bé acompanyada de la opció *-a*, que indica que guardarà el resultat un *array*, el nom d'aquest *array* be donat pel següent paràmetre, en l'exemple mostrat és dirà *net_array*, els *<<<* que venen a continuació indiquen a la comanda anterior que el seu input serà un *string*, i per últim *\$result*, és el *string* amb el que alimentem la comanda *read*.

Per tal de poder connectar-se amb els *slaves* sense intervenció de l'usuari, s'instal·la *sshpass* juntament amb *openssh-server* si fa falta, *sshpass* permet fer servir una contrasenya per connectar-se a un servidor SSH dins d'uns script, de forma segura. Just després és crea una clau asimètrica de tipus RSA que serà la que es faci servir amb els *slaves*, però com que es comparteix el directori */home* del *master* amb tots els *slaves*, aquests perden l'accés a la clau pública que es copiarà al script *add_slave.sh*, és per aquest motiu que el servidor *master* s'afegeix a si mateix al seu fitxer *known_hosts* del directori */home/odorid* i copia la seva clau pública al seu propi fitxer de *authorized_keys*, d'aquesta manera un cop els *slaves* instal·len el client NFS i s'exporten el directori *home* del *master*, els usuaris o aplicacions continuen podent accedir mitjançant la clau pública en comptes d'haver d'escriure la contrasenya. És important afegir primer el host al fitxer *known_hosts*, per guardar-se el *fingerprint* i que el primer cop que és vol connectar via SSH no es demani introduir per teclat *yes* o *no*.

²³ Network Time Protocol, protocol per a sincronitzar les hores dels rellotges a través d'internet.

```
su $master_name -c "echo \"\$(ssh-keyscan -H $(hostname))\" >>
$KNOWN_HOSTS"
```

La comanda de més a dalt és l'encarregada d'escanejar el propi host i obtenir el seu fingerprint, per després guardar-ho al seu fitxer `known_hosts`, ubicat a `/home/odroid/.ssh`, mitjançant la redirecció per afegir a fitxer `>>` i no `>` que escriuria el fitxer, esborrant el contingut anterior, també s'ha optat per fer servir la comanda `su odroid -c ""`, executant la comanda entre cometes com si fos l'usuari `odroid` encara que el script estigui obligatòriament executat pel `root`, això s'ha fet per problemes quan aquesta comanda l'executava el propi `root`.

Les configuracions que es fan sobre el servei `sshd` més destacables són:

- No permetre que connexions SSH entrants com a usuari `root`, d'aquesta forma s'eviten molts atacs que arriben contínuament a través d'internet.
- Obligar a que es faci servir el protocol 2 de SSH, ja que es més segur que el primer.
- No permetre contrasenyes buides, per evitar atacs.
- Habilitar el forwarding de X11, per si es vol executar software amb interfície gràfica a través de SSH, el client que es vulgui connectar haurà de introduir el paràmetre `-X` a la seva comanda `ssh`.
- No mostrar el contingut de `/etc/motd`, ja que podria donar pistes a possibles atacants.

Totes aquestes configuracions i més, es fan sobre el fitxer `/etc/ssh/sshd_config`.

Per a que les configuracions tinguin efecte, s'ha de reiniciar el dimoni `sshd` amb `systemctl restart sshd`.

Per alliberar memòria de la targeta EMMC, s'esborren alguns dels programes que ocupen que no es fan servir com són `libreoffice`, `thunderbird`, `pacman`, `transmission` i tota la interfície gràfica `Mate`, per més tard ser substituïda per `Xfce`, una de més lleugera pel que fa el consum de recursos, la comanda de desinstal·lació és `apt-get remove --purge libreoffice* thunderbird pacman transmission* mate-* -y`, el paràmetre `remove` indica que el software que ve a continuació s'ha de desinstal·lar, `--purge` indica que a més s'esborrin tots els seus fitxers de configuració i `-y` automatitza la resposta a `yes`, a les preguntes que fa la comanda `apt-get`.

Abans d'instal·lar el servidor VNC, s'instal·la l'entorn gràfic pel sistema operatiu anomenat `Xfce`, caracteritzant-se per ser molt lleuger, és a dir té un baix ús de recursos del sistema, després s'instal·la el servidor VNC anomenat `tightvncserver[21]`.

Totes les instal·lacions fan servir la comanda `apt-get install nom_software -y`, on el paràmetre `-y` fa la mateixa funció que en el cas de les desinstal·lacions. A més es configura el script de inicialització del servidor VNC per a que faci servir l'entorn gràfic `Xfce`, aquest script s'anomena `xstartup` i es troba al directori `.vnc` ubicat al directori `home` de l'usuari, en aquest cas `/home/odroid`.

Per a que tots els nodes del clúster puguin accedir als fitxers que els usuaris volen executar, es comparteix el directori `home`, per a que això sigui possible s'ha fet servir NFS, ja explicat a l'apartat 2.1.2.3. Aquest software s'instal·la de forma normal, amb la comanda `apt-get install nfs-kernel-server -y` i no necessita de cap configuració, un cop instal·lada ja es poden compartir directoris per la xarxa interna, mitjançant el fitxer `/etc/exports`.

Per compartir un directori és necessari indicar la xarxa en la que s'està compartint, mitjançant la IP de la xarxa i la seva mascara en notació CIDR, aquest tipus de notació mostra la mascara de xarxa en un sol valor enter acompanyat de una / al davant, es calcula comptant el nombre de 1 que es troben al passar totes les xifres de la mascara en format decimal a binary.

El format utilitzat al fitxer `/etc/exports` és el següent:

```
/home ${ip_net}${mask_cidr}(rw,no_root_squash,no_subtree_check)
```

El primer paràmetre que trobem, fa referencia al directori que és vol compartir, el segon a la xarxa on es vol compartir, per exemple `172.16.0.1/24`, el següents són configuracions específiques del directori que es vol compartir, per exemple `rw` fa referencia a que es permetrà llegir i escriure sobre aquest.

Després de NFS s'instal·la Munge, sent necessari pel normal funcionament de Slurm.

Un cop instal·lat el paquet *munge*, s'habilita el seu dimoni i es crea una nova clau per a compartir entre tots els dimonis *munge* de la xarxa, amb l'objectiu de que els nodes és puguin autenticar entre ells dintre de la xarxa. La creació d'aquesta clau es fa mitjançant la comanda `/usr/sbin/create-munge-key -f`, on el paràmetre `-f` força a crear una nova clau encara que existeixi una. S'ha decidit copiar aquesta clau al directori `/home`, els motius, s'expliquen a l'apartat 4.2.1.2 de la pàgina 36.

4.1.1.2 install_slurm.sh

Aquest script té com objectiu instal·lar i configurar Slurm al servidor paral·lel distribuït. El software s'instal·la amb *apt-get install*, com tots els demés, encara que requereix de configuracions addicionals per poder funcionar, a més de tindre instal·lat Munge i NFS amb el directori `/home` compartit. També es requereix que tots els nodes tinguin comparteixin l'usuari o usuaris amb els que s'executaran programes al clúster, havent de tenir el mateix UID, GID i la mateixa hora.

Per a que Slurm pugui treballar amb *cgroups* és necessari crear un directori anomenat *cgroup*, al mateix directori on es troba el fitxer `slurm.conf`[7] , en el nostre cas `/etc/slurm-lnl` i fer les següent:

- Copiar el fitxer `cgroup[20].release_common` ubicat a `/usr/share/doc/slurmd/examples` al directori *cgroup* creat anteriorment.
- Crear 3 enllaços simbòlics del fitxer copiat cap al mateix directori *cgroup*, amb els següents noms: `release_devices`, `release_cpuset` i `release_freezer`.
- Crear un fitxer al directori *cgroup* amb el següent nom `allowed_devices.conf` amb el següent contingut:

```
/dev/null
/dev/urandom
/dev/zero
/dev/cpu/*/.*
/dev/pts/.*
```

- Per últim s'ha de crear el seu fitxer de configuració al mateix directori que `slurm.conf` i amb el nom de `cgroup.conf`, per aquest projecte se l'hi ha donat la següent configuració:

```

CgroupAutomount=yes
CgroupReleaseAgentDir="\${SLURM_ETC}/cgroup\"
ConstrainCores=yes
TaskAffinity=yes
ConstrainDevices=yes
AllowedDevicesFile="\${SLURM_ETC}/allowed_devices.conf\"
ConstrainRAMSpace=no

```

Després s'ha de crear el fitxer de configuració de *slurm*, anomenar *slurm.conf* i donar-li la configuració desitjada, tots els dimonis *slurmd* i el *slurmctld*, fan servir aquest fitxer.

Per assegurar que tots els fitxers siguin propietat de l'usuari *slurm* s'executa la següent comanda: `chown -R slurm: "$SLURM_ETC"` on el parametre `-R` fa que sigui recursiva i *slurm:* indica que l'usuari i grup propietaris seran *slurm*.

Per finalitzar s'exporta el directori de configuració de *slurm* a tots els nodes de la xarxa interna, i s'habilita i engega el dimoni *slurmctld*.

4.1.1.3 set_vnc_password.sh

Aquest és un script molt senzill amb l'únic objectiu d'afegir una contrasenya al servidor VNC.

Com que el procés d'afegir la contrasenya necessita de la interacció amb l'usuari, per automatitzar-lo s'ha decidit utilitzar un *software* anomenat *expect*, el seu funcionament és molt senzill, només se l'hi han d'indicar les preguntes que farà el programa amb la paraula *expect* i les respostes que ha de donar amb la paraula *send*. Al començament s'ha de fer un *spawn* del programa amb el que interactuarà *expect*, en el nostre cas `/usr/bin/vncpasswd`, el *software* encarregat de modificar la contrasenya del servidor VNC.

```

/usr/bin/expect <<EOF
spawn "$prog"
expect "Password:"
send "$pass\r"
expect "Verify:"
send "$pass\r"
expect "Would you like to enter a view-only password (y/n)?"
send "n\r"
expect eof
exit
EOF

```

4.1.2 Localització

4.1.2.1 locale.sh

Aquest script té el propòsit de fer la localització dels sistemes operatiu que formen el clúster, és a dir canviar l'idioma amb el que el sistema mostra a l'usuari el text, en cas de que tingui la traducció i també canviar el *layout* del teclat que agafarà com a *input* quan l'usuari escrigui, s'ha de remarcar que el canvi de teclat només te efecte quan s'escriu directament amb un teclat connectat a la maquina física, ja que quan es connecti per SSH agafarà el *layout* que estigui configurat a la maquina de l'usuari i el servidor VNC instal·lat no fa servir el teclat de X11, per qüestions de temps no s'ha pogut canviar, però s'hauria de passar de *tightvncserver* a *x11vnc*, sent un problema si es vol que els usuaris tinguin aquesta configuració de forma automàtica.

En el cas de la funció *set_language*, aquesta comprova que l'idioma introduït al fitxer de configuració sigui correcte, consultant que es trobi al fitxer */usr/share/i18n/SUPPORTED* mitjançant la comanda *grep ^"\$locale".UTF-8*, aquesta comanda serveix per buscar un text a un fitxer, el caràcter *^* indica que la coincidència ha de ser al inici de la línia, s'ha decidit només treballar amb UTF-8, per això es busquen les que tenen codificació UTF-8.

locale-gen "\$locale".utf8 instal·la o genera el nou idioma al sistema operatiu. Per indicar al sistema operatiu que ha de fer servir aquest nou idioma s'utilitza la comanda *localectl set-locale LANG="\$locale".UTF-8 LANGUAGE="\$locale".UTF-8:"\$(echo \$locale | cut -d_ -f1)"* el que fa es modificar el fitxer */etc/default/locale* amb les variables LANG i LANGUAGE que farà servir el sistema per saber quin idioma ha de fer servir.

Per a que el sistema rellegeixi el fitxer */etc/default/locale* i apliqui el nou idioma, es fa servir la comanda *source /etc/default/locale*, aquesta mateixa comanda s'afegeix als fitxers */etc/profile*, */etc/bash.bashrc*, */root/.profile* i */root/.bashrc*, els *profile* s'apliquen quan un usuari inicia sessió al sistema i els *bashrc* quan es crea un nou terminal, els que s'ubiquen al fitxer */etc* s'apliquen a tots els usuaris menys a *root*, per això s'ha afegit també al directori *root*.

Per finalitzar es tradueix tot el software com per exemple els navegadors web com Chrome i Firefox, tots aquests paquets a traduir es troben mitjançant la comanda *check-language-support -l "\$locale"*.

En el cas del canvi de teclat s'ha tingut en compte que s'estigui fent servir un entorn gràfic, per fer això és consulta la variable d'entorn *DISPLAY*. Per fer el canvi de teclat s'executa la comanda *setxkbmap -layout \$layout -variant \$variant*, on el primer paràmetre indica quin *layout* es farà servir i el segon quina variant d'aquest, per exemple espanyol com a *layout* i català com la seva variant.

Per a que els canvis de teclat sempre tinguin efecte, al igual que en el canvi d'idioma, s'afegeix la comanda *setxkbmap* explicada anteriorment als fitxers */etc/profile*, */etc/bash.bashrc*, */root/.profile* i */root/.bashrc*.

4.1.3 Xarxa

4.1.3.1 conf_network_master.sh

Aquest script té com a objectiu aplicar algunes de les configuracions de xarxa sobre el servidor master, necessàries per a que el servidor detecti les dues interfícies de xarxa que tindrà, a la interfície de xarxa que connecta amb la xarxa interna se l'hi assignarà una IP fixa,

dintre del rang escollit per l'usuari, mentre que a la interfície que connecta amb internet se l'hi assignarà dinàmicament una direcció IP, mitjançant el servidor DHCP del *router* de l'usuari. També s'habilita el forwarding/encaminament, això el que fa és permetre que les connexions de la xarxa interna passin a través del node *master* i puguin sortir a internet, amb l'objectiu de que els *slaves* puguin instal·lar software i actualitzar-se.

A més per a que el node *master* es 'trobi' a si mateix, es a dir, el seu nom de host, es modifica el fitxer */etc/hosts* afegint la paraula *master* juntament amb la IP de loopback (127.0.0.1). Per poder fer aquesta modificació s'ha fet servir la comanda *sed -i 's/^"\$line"*/"\$line" "\$host"/g' /etc/hosts*, molt resumidament ja que la comanda *sed* és una comanda molt complexa i amb la que es poden fer moltes coses, en aquest cas el que fa es concatenar el contingut original de la línia escollida del fitxer, amb la paraula *master*:

```
line=127.0.0.1 localhost
host=master
127.0.0.1 localhost → 127.0.0.1 localhost master
```

Després mira quina de les dues interfícies de xarxa és la que te connexió amb internet i quina és la que connecta amb la xarxa interna, mitjançant una funció anomenada *check_interfaces* allotjada al script *network_lib.sh*. Això ho fa perquè és necessari poder diferenciar-les, ja que per exemple ha de poder fixar la IP i mascara donades per l'usuari a la interfície que connecta amb la xarxa interna.

Per poder fer aquesta configuració s'ha de modificar el fitxer */etc/network/interfaces* mitjançant la comanda *echo*, amb la següent configuració.

```
echo "
auto $lan_interface
iface $lan_interface inet static
address $ip
netmask ${mask//:/.}" > /etc/network/interfaces
```

També s'afegeixen els noms que tinguin les interfícies de xarxa al fitxer */run/network/ifstate*, per poder reiniciar-les i en el cas de la interfície de la xarxa interna, que pugui obtenir la configuració que se l'hi ha donat, mitjançant les comandes *ifup* i *ifdown*.

Per habilitar el *forwarding* de paquets de forma permanent, es descomenta la línia *net.ipv4.ip_forward=1* mitjançant també la comanda *sed*. Per a que el servidor sen adoni d'aquest canvi s'ha d'executar la comanda *sysctl -p*.

Per últim, es crida al script *iptables.sh*, per a que configuri el tallafocs del sistema operatiu, i es guarda la seva configuració als fitxers */etc/iptables/rules.v4* i */etc/iptables/rules.v6*, per a que el servei *netfilter-persistent*, pugui restablir-les, ja que si no es perdrien al reiniciar el sistema.

Aquest script retorna la IP, mascara, interfície de internet i interfície de la xarxa interna per a que el script *init_master.sh*, que és qui l'ha cridat, pugui disposar dels seus valors i fer-los servir mes endavant.

4.1.3.2 *network_lib.sh*

Aquest script només te funcions i no executa codi per si mateix, si no que és utilitzat per altres scripts com si es tractes d'una llibreria de funcions.

Una llibreria de funcions és un conjunt de funcions que normalment estan agrupades segons els objectius d'aquestes. En aquest cas totes tenen relació amb la xarxa, com treballar amb la adreça IP, la mascara de xarxa o les interfícies de xarxa. Aquestes funcions s'han implementat amb les necessitats sorgides a l'implementar el *software*.

La llibreria consta de 8 funcions dos d'elles, *check_ip* i *check_mask*, d'ús intern, ja que les criden altres funcions del mateix script.

Les funcions *check_ip* i *check_mask* tenen com a objectiu comprovar que la IP en el primer cas, i la mascara en el segon, estiguin en un format correcte, tenint en compte que la IP ha de tindre 4 xifres separades per punts i a d'estar dintre del rang de IPs privades. Per la mascara de xarxa es té en compte que tingui 4 xifres separades per dos punts, i que cada una de les xifres estigui dintre del rang de valors 0 a 255.

La interfície gràfica ha estat programada tenint en compte de que l'usuari només pugui introduir IP i mascara de xarxa valides. Per aquest motiu es recomana fer servir la interfície gràfica, i no modificar manualment el fitxer *odroid_cluster.conf*, a no ser que es tinguin coneixements avançats.

Classe	Inici del rang	Final del rang
A	10.0.0.0	10.255.255.255
B	172.16.0.0	172.31.255.255
C	192.168.0.0	192.168.255.255

Taula 3: Classes de IP privada i el seus respectius rangs.

La funció *check_interfaces* localitza el nom de totes les interfícies de xarxa i comprova quina pertany a la xarxa local i quina a internet.

Per trobar els noms de les interfícies de xarxa es consulta el fitxer */proc/net/dev*, i es fica en un format fàcil de recórrer dins d'un bucle, amb la comanda *sed*, no es té en compte la interfície *lo* ja que és una interfície de xarxa especial que fa referència a *loclahost*, i no comunica amb cap xarxa, si no, amb si mateix.

Per cada una de les interfícies de xarxa que es trobin es farà un *ping* a una IP pública, es a dir, ubicada a fora de la xarxa interna. Si aquesta màquina respon al *ping* voldrà dir que aquella interfície és la que connecta a internet, si no respon, és perquè forma part de la xarxa interna. En cas de que hagués una tercera interfície de xarxa el software podria no funcionar correctament, ja que no podria distingir per exemple quina de les dos xarxes internes és la que connecta o connectarà amb els nodes *slaves*.

```
ping 8.8.8.8 -I $nic -w2 2> /dev/null | grep "received" | cut -d " " -f4
```

A dalt es pot veure la comanda *ping* que s'ha fet servir per classificar les interfícies de xarxa, on el primer valor numèric separat per punts és la IP publica objectiu del *ping*, el paràmetre *-I* indica la targeta de xarxa per on s'enviarà el *ping*, el paràmetre *-w* serveix per limitar el temps en el que s'executarà el *ping* a N segons en aquest cas 2.

get_ip_of_nic és una funció que serveix per retornar l'adreça IP assignada a una targeta de xarxa, que rep per paràmetre.

mask_to_cidr és una altra funció que canvia el format de la mascara, passant de 4 xifres separades per dos punts a una / seguit d'una única xifra, tots amb valors enters.

Com per exemple:

$255:255:255:0 \rightarrow /24$

Per fer aquest canvi el que es fa és passar cada una de les xifres en decimal a binary i comptar el nombre de uns totals que formen la màscara. S'utilitza la comanda `obase=2;${i}|bc`, fent la modificació de base. Per últim es compten tots els uns amb la comanda `awk -F "1" '{print NF-1}'`.

Les funcions `calculate_network_ip`, `calculate_first_ip` i `calculate_last_ip` fan exactament el mateix a excepció de una línia, que és la que marca el resultat a retornar.

Totes tres reben per paràmetre una IP i una mascara, primer comproven que el format de la IP i la mascara sigui correcte, amb les funcions `check_ip` i `check_masck` respectivament,

Després passen la IP i la mascara a 4 variables, una per cada xifra que formen la IP i la mascara. El funcionament d'aquestes comandes es el mateix que l'explicat a l'apartat 4.1.1.1 `init_master.sh`, amb l'única diferència de que en comptes de guardar els valors a un vector es guarden a 4 variables amb el paràmetre `-r`.

```
IFS=. read -r i1 i2 i3 i4 <<< "$ip"
IFS=: read -r m1 m2 m3 m4 <<< "$mask"
```

Les tres comandes apliquen una mascara de bit de tipus AND entre la IP i la màscara, amb petites diferències com incrementar en 1 l'última xifra per a `calculate_first_ip` i ja sent més complexa de `calculate_last_ip` fent servir de més a més una màscara OR:

- `calculate_network_ip`:

```
result=$(echo "$((i1 & m1)).$((i2 & m2)).$((i3 & m3)).$((i4 & m4))")
```

- `calculate_first_ip`:

```
result=$(echo "$((i1 & m1)).$((i2 & m2)).$((i3 & m3)).$(((i4 & m4)+1))")
```

- `calculate_last_ip`:

```
result=$(echo "$((i1 & m1 | 255-m1)).$((i2 & m2 | 255-m2)).$((i3 & m3 | 255-m3)).$(((i4 & m4 | 255-m4)-1))")
```

4.1.3.3 dhcp_script.sh

Aquest script s'executa cada cop el servidor DHCP de `dnsmasq` rep una petició per part d'un client d la xarxa interna. `Dnsmasq` s'encarrega de passar-li 4 paràmetres:

1. Acció:
 - `add`: Nou client DHCP.
 - `old`: Client DHCP conegut.
 - `del`: Client DHCP esborrat.
2. Adreça MAC (física) del client.
3. Adreça IP del client.
4. Nom del client.

En aquest script en concret només s'aprofita el paràmetre `add`, ja que el seu objectiu es executar codi quan es rep un nou client DHCP, per inicialitzar el node.

add_odroid és la funció encarregada de guardar la MAC, el nom de host assignat al nou node i la seva IP, per calcular el nom de host es consulta el fitxer *odroid_cluster.conf*, agafant el nom a partir de la paraula clau *HOSTS_NAME*, després s'agafa el número de línies del fitxer *dnsmasq_hosts.conf* ubicat a */etc/dnsmasq.d/* que és una extensió del fitxer de configuració de dnsmasq, */etc/dnsmasq.conf*. Un cop es té el número de línies, s'incrementa en 1, és a dir, el nom de host és incremental i s'inicia en 1.

Exemple:

Si *dnsmasq_hosts.conf* té 42 línies, i és vol afegir un nou node, el nom de host serà

$$odroid + 42 + 1$$

Quedant en:

$$odroid43$$

S'agafa el numero de línies de *dnsmasq_hosts.conf* perquè en aquest fitxer és guarden tots els nodes de la xarxa interna, als que se l'hi dona una IP fixa i el nom de host calculat, amb l'adreça MAC del node que és vol afegir.

El format que segueix el fitxer *dnsmasq_hosts.conf* comença sempre per la paraula *dhcp-host=* i segueix per la MAC, el nom de host i la IP, aquests últims tres valors separats per comes.

$$dhcp-host=MAC,HostName,IP$$

A més, la funció *add_odroid*, incrementa en 1 el nombre de hosts guardats al fitxer de configuració de slurm, ubicat a */etc/slurm-llnl/slurm.conf*, per a que els dimonis de *slurm* tinguin coneixement de que el nombre de nodes s'ha incrementat. És necessari reiniciar el dimoni *slurmctld*, a més dels dimonis *slurmd* ubicats als *slaves*, encara que aquesta funció només reinicia *slurmctld*, s'ha implementat un mecanisme per a que els *slurmd* es reiniciïn automàticament quan es modifica *slurm.conf*.

Com que */etc/dnsmasq.d/dnsmasq_hosts.conf* és una extensió de */etc/dnsmasq.conf*, el servei de *dnsmasq*, només llegirà els canvis fets a */etc/dnsmasq.d/dnsmasq_hosts.conf* un cop es reiniciï, mitjançant la comanda *systemctl restart dnsmasq* o reiniciant el sistema, això fa que s'hagi de guardar el nou host a un altre fitxer de forma temporal aquest fitxer s'anomena *tmp_hosts* i s'ubica a */etc/hosts.d/*, els hosts és guarden amb el format IP i nom de host.

Per finalitzar aquesta funció s'inicia en *background* i amb la comanda *nohup* el script *add_slave.sh*

```
nohup      "${scripts_path}/add_slave.sh"      "${host_name}${num_line}"
"${get_sleep_time}" >>
/var/log/odroid_cluster/add_slave_"${host_name}${num_line}".out 2>&1 &
```

S'executa en *background* per que si no el *dhcp_script.sh* seria bloquejant, i no és podrien atendre noves peticions DHCP i per tant inicialitzar nous nodes, fins que acabes la inicialització del node actual.

Es fa servir la comanda *nohup*, perquè si no, encara que s'executi *add_slave.sh* en *background*, al terminar l'execució del seu script pare (*dhcp_script.sh*) també s'acabaria immediatament l'execució actual i en *background* de *add_slave.sh*.

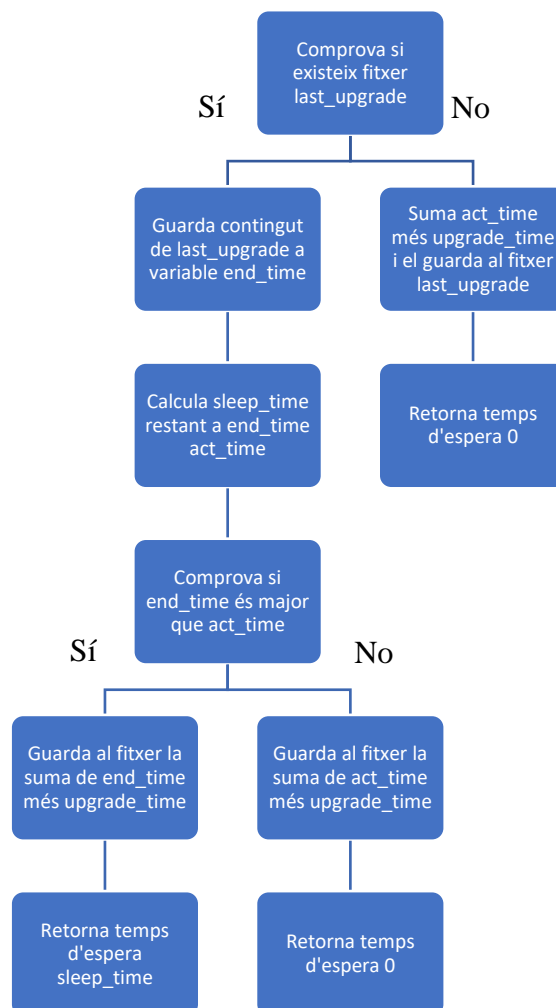
L'altra funció important que guarda aquest fitxer és *get_sleep_time*, aquesta funció calcula els temps d'espera entre *upgrades*, fent la funció d'un planificador molt simple, s'ha

decidit implementar aquest temps d'espera perquè si es connecta un grup de nodes per ser inicialitzats al mateix temps, i es decideix activar la opció de que facin *upgrade* després de ser inicialitzats, en un moment donat tots els nodes començaran descarregar actualitzacions al mateix temps, ocupant tot l'ample de banda de la xarxa i saturant-la, a més de que es podria donar el cas de que deixessin bloquejat el servei de *dnsmasq*.

La funció *sleep_time* fa servir tres elements importants per calcular el temps d'espera entre un *upgrade* i el següent.

- *upgrade_time*: Temps que ha decidit l'usuari que ha d'haver entre una actualització i la següent, guardat al fitxer *odroid_cluster.conf*.
- *act_time*: Temps actual en minuts, des-de la mitjanit UTC del 1 de gener de 1970 fins el moment actual. Aquest és un valor estàndard a molts sistemes operatius per comptabilitzar el temps transcorregut i s'anomena *Epoch time* o *POSIX time*.
- *last_upgrade*: Fitxer que serveix per guardar el moment en el que finalitzarà o ha finalitzat, la última actualització d'un nou node inicialitzat pel *dhcp_script.sh*. Es a dir, només guarda un valor enter de *Epoch time* en minuts.

Ja explicats aquests tres valors, el procediment per calcular el temps d'espera o *sleep* és:



4.1.3.4 iptables.sh

El script `iptables.sh` s'encarrega de configurar el tallafocs, configurant les polítiques i regles que farà servir per filtrar els paquets que passin a través d'ell. Com ja s'ha comentat a l'apartat 3.2 al servidor *master* fa al mateix temps de Firewall/Tallafocs, per això s'instal·la *iptables* al *master*.

Per incrementar la seguretat s'han decidit aplicar unes polítiques restrictives, es a dir, si no és diu el contrari tots els paquets que intenten passar per o entre el *master*, es llencen.

Algunes de les regles que s'han fet servir escrites en paraules són:

- Qualsevol paquet pot entrar o sortir per la interfície de *loopback*, això és degut a que forma part de les comunicacions del propi *master* entre sí.
- Tots els paquets que pertanyin a connexions establertes se'ls permetrà sortir, entrar per les dos interfícies de xarxa, i travessar el *master*.
- S'habilitarà el *postrouting*, a la taula nat de iptables, amb l'objectiu de que els paquets que surten de la xarxa interna cap a internet, sàpiguen a on han de tornar.

Per facilitar l'explicació la resta es mostrarà en una taula:

Servei	Port	Interfície	Entrada	Sortida	Encaminament
SSH	22	Internet	Sí	No	No
SSH	22	LAN	Sí	Sí	No
DNS	53	Internet	No	Sí	No
DNS	53	LAN	Sí	Sí	No
DHCP	67	Internet	No	No	No
DHCP	67	LAN	Sí	No	No
DHCP	68	Internet	No	Sí	No
DHCP	68	LAN	No	Sí	No
HTTP	80	Internet	No	Sí	LAN → Internet
HTTP	80	LAN	No	No	LAN → Internet
HTTPS	443	Internet	Sí	Sí	LAN → Internet
HTTPS	443	LAN	No	Sí	LAN → Internet
Node.js	3000	Internet	Sí	No	No
Node.js	3000	LAN	No	No	No
VNC	5901	Internet	Sí	No	No
VNC	5901	LAN	No	No	No
SLURM	6817	Internet	No	No	No
SLURM	6817	LAN	Sí	No	No
SLURM	6818	Internet	No	No	No
SLURM	6818	LAN	No	Sí	No

PING	ICMP 8	Internet	No	Sí	→ Internet
PING	ICMP 8	LAN	Sí	Sí	→ Internet
NFS	2049	Internet	No	No	No
NFS	2049	LAN	No	Sí	No
NTP	123	Internet	No	Sí	→ Internet
NTP	123	LAN	No	No	→ Internet
apt-key	11371	Internet	Sí	No	→ Internet
apt-key	11371	LAN	No	No	→ Internet

Taula 4: Configuració regles *iptables*.

Degut a que la comanda *srun* de *slurm*, que serveix per executar programes al clúster, fa servir diversos ports dintre del protocol TCP, al final s'han permet totes les connexions d'entrada, per la xarxa interna, si no, aquesta comanda quedava bloquejada. També s'han permet les connexions del protocol ICMP²⁴ dintre la xarxa interna.

Quan és diu travessar o passar a través de, es fa referència a l'encaminament de paquets o forwarding.

4.2 Scripts *slaves*

4.2.1 Generals

4.2.1.1 *add_slave.sh*

Aquest script es té com a únic propòsit, facilitar l'accés mitjançant clau pública al nou node, copiar els scripts necessaris per a inicialitzar el node com a *slave*, sent *init_slave.sh* i *locale.sh* i per últim executar el script *init_slave.sh* que s'ha copiat al node, mitjançant SSH.

Comandes importants:

Quan es vol accedir a un computador per primer cop via SSH, és pregunta a l'usuari si està segur de continuar amb la connexió, per evitar aquesta pregunta dins d'un script, és pot escanejar la clau SSH del host objectiu, per emmagatzemar-la al fitxer de *known_hosts*. Aquesta comanda s'executa com al usuari *odroid*, o el que estigui assignat al fitxer de configuració, el paràmetre *-c* de *su* serveix per indicar que es vol executar la comanda que està entre cometes, el paràmetre *-H* de *ssh-keyscan* serveix per indicar el host que es vol escanejar, les *>>* serveixen per afegir la sortida del la comanda anterior al fitxer, si només es fiqués una *>* s'esborrarien tots el hosts que s'han afegit amb anterioritat.

```
su $user_name -c "echo \"$(ssh-keyscan -H $host)\" >> $KNOWN_HOSTS"
```

Un cop guardat el *fingerprint* del host al que es vol connectar, es procedeix a copiar la clau pública RSA, per a que és pugui accedir sense escriure contrasenya, mitjançant la comanda *ssh-copy-id*. La funció de *su* és la mateixa que al cas anterior i s'afegeix *sshpass*

²⁴ Internet Control Message Protocol, protocol per a enviar missatges.

que serveix per a que escrigui per l'usuari la contrasenya, en aquest cas la que ve per defecte a les imatges de ODROID, el paràmetre `-p` serveix per indicar la contrasenya a utilitzar, el paràmetre `-i` de la comanda `ssh-copy-id` indica quina clau es vol copiar, podria ser que tinguéssim múltiples claus generades i només volem copiar una en concret. Per indicar a qui es vol copiar s'ha de descriure el nom d'usuari a qui volem enviar la clau pública, en el nostre cas usuari `odroid`, una `@` com a separador i el nom de host o la seva IP.

```
su $user_name -c "sshpass -p $default_password ssh-copy-id -i $KEY_FILE $user_name@$host"
```

Aquestes comandes són molt semblants a les anteriors amb la diferència de que en aquest cas es fica `scp` en comptes de `ssh-copy-id`, després de `scp` s'indica el fitxer que es vol copiar, en aquest cas es fa servir un *path* absolut per assegurar de que trobi el script a copiar. A més, després de `user@host` s'afegeixen dos punts i el directori destí del fitxer. Es pot observar que la comanda `sshpass` ha desaparegut, perquè ja no fa falta al haver copiat la clau pública.

```
su $user_name -c "scp ${scripts_path}/locale.sh ${user_name}@${host}:Documents"
su $user_name -c "scp ${scripts_path}/init_slave.sh ${user_name}@${host}:Documents"
```

Per executar el script ja copiat es fa mitjançant la comanda SSH, com que s'ha d'executar mitjançant usuari `root`, s'ha fet servir la comanda `sudo` amb el paràmetre `-S`, i que recull la contrasenya que l'hi envien per l'*output* de la comanda anterior, amb l'objectiu d'accedir amb privilegis de *super* usuari.

```
su $user_name -c "ssh -t ${user_name}@${host} \"echo ${default_password} | sudo -S ~/Documents/init_slave.sh ${master_ip} ${upgrade_slave} ${upgrade_time} ${locale} \" >> /var/log/odroid_cluster/init_slave_${host}.out 2>&1"
```

Amb això fet ja s'estarà instal·lant i configurant tot el *software* necessari al nou slave.

4.2.1.2 init_slave.sh

Aquest script és l'encarregat de inicialitzar els slaves, instal·lant i configurant tot el software necessari, fa ús del script `locale.sh` per configurar l'idioma i el layout del teclat.

La funció `add_slurm` instal·la el software *slurm* mitjançant la comanda ja comentada `apt install slurm-wlm -y`, munta la partició que ha compartit el node màster mitjançant NFS i crea el directori `/var/spool/slurmd`, ja que el servei *slurmd* necessita treballar sobre el directori `/var/spool` però no l'hi podem ficar com a propietari de *spool* a l'usuari *slurm*. Per últim s'habilita el dimoni *slurmd*, necessari als *slaves*.

La funció `add_slurm_watcher` l'únic que fa és crear un nou servei de *systemd* anomenat *slurm_watcher*, aquest servei té com a objectiu vigilar el fitxer `slurm.conf`, ubicat a `/etc/slurm-llnl` i quan aquest es modifica, reiniciar el servei *slurmd*, per a que detecti els canvis realitzats. D'aquesta forma no s'ha d'accedir a tots els *slaves* si es vol modificar el fitxer de configuració de *slurm*.

Fitxer `slurm_watcher.service`:

```
[Unit]
Description=Restart slurmd if slurm.conf is modified.
After=network.target

[Service]
Type=oneshot
ExecStart=$(which systemctl) restart slurmd.service

[Install]
WantedBy=multi-user.target
```

Fitxer `slurm_watcher.path`:

```
[Path]
PathModified=${SLURM_ETC}/slurm.conf

[Install]
WantedBy=multi-user.target
```

La constant `SLURM_ETC` fa referència a `/etc/slurm-llnl`. Tots dos fitxers s'ubiquen al directori `/etc/systemd/system`.

Per habilitar aquest tipus de dimoni amb més d'un fitxer es pot fer servir la comanda:

```
systemctl enable --now slurm_watcher.{path,service}
```

D'aquesta forma, s'habiliten i inicien els dos alhora. La opció `enable` indica que s'habilitin els dimonis i el paràmetre `--now` serveix per iniciar-los sense haver de fer després la mateixa comanda amb l'opció `start`.

Com que aquest script necessita 4 arguments per poder treballar, primer valida que se l'hi hagin passat.

Per començar fica al servidor *master* com l'únic servidor DNS, el servidor DNS és necessari per resoldre el noms de domini, d'aquesta manera passa de un nom a una direcció IP, entre d'altres. Per exemple quan es fica al navegador *google.com*, el servidor que estigui configurat traduirà aquest domini a una IP, fent que no hakis de tindre en compte les IPs si no que només has de saber un nom de domini. En aquest cas es configura el servidor *master* com a DNS dels *slaves*, introduint la seva IP al fitxer `/etc/resolv.conf`. D'aquesta manera, ja no només es resoldran els noms de domini externs, com *google.com* si no que també els interns, com *master* o *odroid1*, etc.

Com que el fitxer `/etc/resolv.conf`, no és un fitxer normal, si no que és un enllaç a un altre fitxer, fet servir per un dimoni de Ubuntu, si modifiques `resolv.conf` al reiniciar tornarà a la seva configuració original, hi ha solucions com instal·lar un servei anomenat `resolvconf`, que s'encarrega d'estar actualitzant el fitxer `resolv.conf` amb la configuració que se l'hi doni. Com que van haver problemes i conflictes amb aquest dimoni, durant el desenvolupament del projecte, s'ha optat per una opció més senzilla, esborrar el `/etc/resolv.conf` original i tornar-lo a crear amb la paraula `nameserver` i la IP del *master*, per a que no sigui un enllaç. Per a que el dimoni propietari de `resolv.conf` no torni a modificar aquest fitxer, s'ha optat per bloquejar-lo i que sigui immutable, a no ser que es desbloquegi, amb al comanda `chattr`

+i */etc/resolv.conf* el paràmetre +i serveix per modificar un *flag* que no deixa modificar el fitxer, la contrapart es el paràmetre -i.

A l'igual que al *master* es configura la interfície de xarxa mitjançant el fitxer */etc/network/interfaces*, encara que en aquest cas és vol que se l'hi assigni la IP per DHCP.

Després trobem coses molt semblants a *init_master.sh*, com la modificació de la zona horària, la instal·lació i habilitació del servei *chrony*, la resolució del problema amb les claus del servidor *keyserver.ubuntu.com* i l'actualització dels repositoris del sistema mitjançant la comanda *apt update*.

A diferència del *master* s'instal·la el software *sysstat*, a petició de Joan Jara Bosch, que participa en un projecte continuació d'aquest. Aquest software és un recull d'utilitats per monitoritzar els nodes, com per exemple la comanda *iostat*, que serveix per obtenir informació relacionada amb l'input i l'output del medi d'emmagatzematge físic, com per exemple un disc dur o la memòria EMMC.

Pel que fa el servidor SSH en aquest cas l'únic que es fa és deshabilitar l'accés com a *root*.

També es desinstal·len el mateixos programes que a *init_master.sh*.

S'instal·la el client NFS amb la comanda *apt-get install nfs-common -y*. I s'afegeix el directori */home* compartit pel *master*, requisit per fer funcionar el servidor paral·lel distribuït. Ja que els slaves necessiten poder accedir als fitxers que es volen executar al clúster. Per fer-ho s'afegeix la següent línia al fitxer */etc/fstab*. La variable *master_ip* fa referència a la IP del *master*, els dos punts */home* fan referència al directori dels que comparteix el servidor, i el segon */home* al directori on es vol muntar del *slave*, *nfs* indica el tipus de sistema de fitxers, *rw* que es munta amb lectura i escriptura, *auto* que es munta a l'inici i *_netdev* indica que el sistema de fitxers requereix d'accés a la xarxa.

```
$(master_ip):/home /home nfs rw,auto,_netdev 0 0
```

S'instal·la i habilita el dimoni *Munge*, per a que *slurm* pugui funcionar correctament, com ja s'ha explicat a l'apartat *Munge2.1.3.3* de la pàgina 8. Després es copia la clau necessària, compartida al directori *home* i es reinicia el dimoni de *munge* per a que reconeixi la nova clau. S'ha utilitzat la comanda *dd if=/home/munge.key of=/etc/munge/munge.key* per fer la copia, el paràmetre *if* fa referència al fitxer d'entrada, i *of* al de sortida, encara que es podria haver utilitzat perfectament la comanda *cp -a*. Encara que estigui a */home* no representa cap problema ja que només l'usuari *munge* o el *root*, tenen permisos per llegir o modificar aquest fitxer, i d'aquesta manera va facilitar el disseny del software, encara que, en futures versions es podria mirar de canviar.

Es crida a la funció *add_slurm*, explicada a l'apartat **¡Error! No se encuentra el origen de la referencia.**

En el cas de MPICH s'instal·la amb *apt-get install* i no s'ha de configurar res.

El node *slave* guarda la següent resposta al fitxer *.slave_responses*, ubicat al seu directori *ho*

```
I am $(hostname) I have already installed and configured everythingme.
```

La comanda *hostname* retorna el nom de la màquina o nom de host. Aquest fitxer té un punt al l'inici del nom per indicar que està ocult. Un usuari administrador podria mirar si aquest node a acabat, consultant aquest arxiu o els fitxers de *log*. Aquest fitxer es va crear

amb la intenció de tindre un apartat a la interfície gràfica que el consultes i mostres quins nodes han acabat de instal·lar-se, però no s'ha introduït per falta de temps. És va ficar al directori home de l'usuari *odroid* ja que és un accés fàcil per al node master.

Per últim s'executen les funcions de localització, explicades a l'apartat 4.1.2.1 de la pàgina 28. I és fa un *upgrade* s'hi s'ha ficat la opció *UPGRADE=* a 1 al fitxer de configuració.

5 Avaluació

Per tal d'avaluar la correcta execució del codi s'ha optat per comprovar els outputs que es generen durant l'execució del software, aquests outputs estan dividits en 2 parts, per un costat s'analitza la correcta execució sobre el servidor *master*, i per altra banda s'ha agafat els fitxers de *log* d'un dels *slaves* afegits al clúster. Per a que no ocupi massa pàgines, s'ha esborrat la part on es desinstal·len els programes i la part on s'instal·la el software de monitorització de Joan Jara Bosch, ja que ell ja avalua el correcte funcionament d'aquest, en el seu projecte.

5.1 Servidor master

Al següent tros de codi es pot veure com efectivament es deshabilita i desinstal·la el software de Ubuntu, *unattended-upgrades*:

```
Synchronizing state of unattended-upgrades.service with SysV service script with
/lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable unattended-upgrades
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libdrm-freedreno1 libllvm6.0
Use 'sudo apt autoremove' to remove them.
The following packages will be REMOVED:
  unattended-upgrades
0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.
After this operation, 393 kB disk space will be freed.
(Reading database ... 153565 files and directories currently installed.)
Removing unattended-upgrades (1.1ubuntu1.18.04.11) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
```

En aquesta part es pot veure com es soluciona el problema amb la clau faltant per actualitzar, i com s'actualitza el sistema:

```
Executing: /tmp/apt-key-gpghome.Zetrmggzt/gpg.1.sh -v --keyserver
keyserver.ubuntu.com --recv-keys 5360FB9DAB19BAC9
gpg: no running dirmngr - starting '/usr/bin/dirmngr'
gpg: waiting for the dirmngr to come up ... (5s)
gpg: waiting for the dirmngr to come up ... (4s)
gpg: connection to dirmngr established
gpg: data source: http://162.213.33.9:11371
gpg: pub      rsa4096/4047B34823AE12F8  2015-01-07      Mauro  Ribeiro  (mdrjr)
<mauro.ribeiro@hardkernel.com>
gpg: key 4047B34823AE12F8: "Mauro Ribeiro (mdrjr) <mauro.ribeiro@hardkernel.com>" 1 new
signature
gpg: no running gpg-agent - starting '/usr/bin/gpg-agent'
gpg: waiting for the agent to come up ... (5s)
```

```

gpg: waiting for the agent to come up ... (4s)
gpg: connection to agent established
gpg: Total number processed: 1
gpg:          new signatures: 1
Hit:1 http://ports.ubuntu.com/ubuntu-ports bionic InRelease
Get:2 http://ports.ubuntu.com/ubuntu-ports bionic-updates InRelease [88.7 kB]
Get:3 http://archive.canonical.com/ubuntu bionic InRelease [10.2 kB]
Get:4 http://ports.ubuntu.com/ubuntu-ports bionic-backports InRelease [74.6 kB]
Get:5 http://ports.ubuntu.com/ubuntu-ports bionic-security InRelease [88.7 kB]
Get:6 http://deb.odroid.in/c2 bionic InRelease [2406 B]
Get:7 http://archive.canonical.com/ubuntu bionic/partner Sources [1908 B]
Get:8 http://archive.canonical.com/ubuntu bionic/partner Translation-en [1332 B]
Get:9 http://ports.ubuntu.com/ubuntu-ports bionic-updates/restricted Sources [15.9 kB]
Get:10 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main Sources [496 kB]
Get:11 http://ports.ubuntu.com/ubuntu-ports bionic-updates/universe Sources [434 kB]
Get:12 http://ports.ubuntu.com/ubuntu-ports bionic-updates/multiverse Sources [10.4
kB]
Get:13 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 Packages [1120
kB]
Get:14 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main Translation-en [367
kB]
Get:15 http://ports.ubuntu.com/ubuntu-ports bionic-updates/restricted arm64 Packages
[2088 B]
Get:16 http://ports.ubuntu.com/ubuntu-ports bionic-updates/restricted Translation-en
[24.6 kB]
Get:17 http://ports.ubuntu.com/ubuntu-ports bionic-updates/universe arm64 Packages
[1480 kB]
Get:18 http://ports.ubuntu.com/ubuntu-ports bionic-updates/universe Translation-en
[354 kB]
Get:19 http://ports.ubuntu.com/ubuntu-ports bionic-updates/multiverse arm64 Packages
[5072 B]
Get:20 http://ports.ubuntu.com/ubuntu-ports bionic-updates/multiverse Translation-en
[6980 B]
Get:21 http://ports.ubuntu.com/ubuntu-ports bionic-backports/universe Sources [5360 B]
Get:22 http://ports.ubuntu.com/ubuntu-ports bionic-backports/main Sources [5440 B]
Get:23 http://ports.ubuntu.com/ubuntu-ports bionic-backports/main arm64 Packages [9992
B]
Get:24 http://ports.ubuntu.com/ubuntu-ports bionic-backports/main Translation-en [4764
B]
Get:25 http://ports.ubuntu.com/ubuntu-ports bionic-backports/universe arm64 Packages
[9932 B]
Get:26 http://ports.ubuntu.com/ubuntu-ports bionic-backports/universe Translation-en
[4588 B]
Get:27 http://ports.ubuntu.com/ubuntu-ports bionic-security/multiverse Sources [4148
B]
Get:28 http://ports.ubuntu.com/ubuntu-ports bionic-security/main Sources [237 kB]
Get:29 http://ports.ubuntu.com/ubuntu-ports bionic-security/restricted Sources [11.4
kB]
Get:30 http://ports.ubuntu.com/ubuntu-ports bionic-security/universe Sources [268 kB]
Get:31 http://ports.ubuntu.com/ubuntu-ports bionic-security/main arm64 Packages [827
kB]

```

```

Get:32 http://ports.ubuntu.com/ubuntu-ports bionic-security/main Translation-en [277
kB]
Get:33 http://ports.ubuntu.com/ubuntu-ports bionic-security/restricted arm64 Packages
[1172 B]
Get:34 http://ports.ubuntu.com/ubuntu-ports bionic-security/restricted Translation-en
[22.1 kB]
Get:35 http://ports.ubuntu.com/ubuntu-ports bionic-security/universe arm64 Packages
[943 kB]
Get:36 http://ports.ubuntu.com/ubuntu-ports bionic-security/universe Translation-en
[242 kB]
Get:37 http://ports.ubuntu.com/ubuntu-ports bionic-security/multiverse arm64 Packages
[2360 B]
Get:38 http://ports.ubuntu.com/ubuntu-ports bionic-security/multiverse Translation-en
[2964 B]
Get:39 http://deb.odroid.in/c2 bionic/main arm64 Packages [16.6 kB]

```

En aquest tros de codi és vàlida la instal·lació necessària per modificar el *layout* del teclat i la seva modificació:

```

Fetched 7478 kB in 5s (1651 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libdrm-freedreno1 libllvm6.0
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  xorgxrdp
Suggested packages:
  guacamole xrdp-pulseaudio-installer
The following NEW packages will be installed:
  xorgxrdp xrdp
0 upgraded, 2 newly installed, 0 to remove and 371 not upgraded.
Need to get 443 kB of archives.
After this operation, 3132 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 xorgxrdp arm64 0.9.5-
2 [68.7 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 xrdp arm64 0.9.5-2
[374 kB]
Fetched 443 kB in 0s (977 kB/s)
Selecting previously unselected package xorgxrdp.
(Reading database ... 153542 files and directories currently installed.)
Preparing to unpack ../xorgxrdp_0.9.5-2_arm64.deb ...
Unpacking xorgxrdp (0.9.5-2) ...
Selecting previously unselected package xrdp.
Preparing to unpack ../xrdp_0.9.5-2_arm64.deb ...
Unpacking xrdp (0.9.5-2) ...
Setting up xrdp (0.9.5-2) ...

```

```

Generating 2048 bit rsa key...

ssl_gen_key_xrdp1 ok

saving to /etc/xrdp/rsakeys.ini

Created symlink /etc/systemd/system/multi-user.target.wants/xrdp-sesman.service →
/lib/systemd/system/xrdp-sesman.service.
Created symlink /etc/systemd/system/multi-user.target.wants/xrdp.service →
/lib/systemd/system/xrdp.service.
Setting up xorgxrdp (0.9.5-2) ...
Processing triggers for systemd (237-3ubuntu10.25) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...

```

En aquesta part del codi es pot observar com s'instal·la el software expect i com demana a l'usuari la contrasenya nova i la modifica:

```

Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
 libdrm-freedreno1 libllvm6.0
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
 libtcl8.6 tcl-expect tcl8.6
Suggested packages:
 tk8.6 tcl-tclreadline
The following NEW packages will be installed:
 expect libtcl8.6 tcl-expect tcl8.6
0 upgraded, 4 newly installed, 0 to remove and 371 not upgraded.
Need to get 1028 kB of archives.
After this operation, 4405 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports bionic/main arm64 libtcl8.6 arm64
8.6.8+dfsg-3 [781 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 tcl-expect arm64
5.45.4-1 [95.9 kB]
Get:3 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 expect arm64 5.45.4-
1 [137 kB]
Get:4 http://ports.ubuntu.com/ubuntu-ports bionic/main arm64 tcl8.6 arm64 8.6.8+dfsg-
3 [14.3 kB]
Fetched 1028 kB in 0s (2234 kB/s)
Selecting previously unselected package libtcl8.6:arm64.
(Reading database ... 153661 files and directories currently installed.)
Preparing to unpack .../libtcl8.6_8.6.8+dfsg-3_arm64.deb ...
Unpacking libtcl8.6:arm64 (8.6.8+dfsg-3) ...
Selecting previously unselected package tcl-expect:arm64.
Preparing to unpack .../tcl-expect_5.45.4-1_arm64.deb ...

```

```

Unpacking tcl-expect:arm64 (5.45.4-1) ...
Selecting previously unselected package expect.
Preparing to unpack .../expect_5.45.4-1_arm64.deb ...
Unpacking expect (5.45.4-1) ...
Selecting previously unselected package tcl8.6.
Preparing to unpack .../tcl8.6_8.6.8+dfsg-3_arm64.deb ...
Unpacking tcl8.6 (8.6.8+dfsg-3) ...
Setting up libtcl8.6:arm64 (8.6.8+dfsg-3) ...
Setting up tcl-expect:arm64 (5.45.4-1) ...
Setting up tcl8.6 (8.6.8+dfsg-3) ...
Setting up expect (5.45.4-1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for menu (2.1.47ubuntu2.1) ...
Enter the new password for the master node:
Re-enter the new password:

```

A la següent secció de codi és pot observar com s'instal·la el software *chrony* i com s'habilita el seu dimoni, encara que surt un missatge d'error, s'ha pogut comprovar que el servei s'inicia correctament i que fa la seva funció actualitzant l'hora, consultant la comanda *date*:

```

Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libdrm-freedreno1 libllvm6.0
Use 'sudo apt autoremove' to remove them.
Suggested packages:
  dnstools
The following NEW packages will be installed:
  chrony
0 upgraded, 1 newly installed, 0 to remove and 371 not upgraded.
Need to get 184 kB of archives.
After this operation, 464 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 chrony arm64 3.2-4ubuntu4.5 [184 kB]
Fetched 184 kB in 0s (594 kB/s)
Selecting previously unselected package chrony.
(Reading database ... 153990 files and directories currently installed.)
Preparing to unpack .../chrony_3.2-4ubuntu4.5_arm64.deb ...
Unpacking chrony (3.2-4ubuntu4.5) ...
Setting up chrony (3.2-4ubuntu4.5) ...
Creating '_chrony' system user/group for the chronyd daemon...

Creating config file /etc/chrony/chrony.conf with new version

Creating config file /etc/chrony/chrony.keys with new version

```

```

Created symlink /etc/systemd/system/chronyd.service →
/lib/systemd/system/chrony.service.
Created symlink /etc/systemd/system/multi-user.target.wants/chrony.service →
/lib/systemd/system/chrony.service.
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for systemd (237-3ubuntu10.25) ...
Failed to enable unit: Refusing to operate on linked unit file chronyd.service

```

En aquesta part de la execució es veu com s'instal·la el servei de netfilter-persistent:

```

Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libdrm-freedreno1 libllvm6.0
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  netfilter-persistent
The following NEW packages will be installed:
  iptables-persistent netfilter-persistent
0 upgraded, 2 newly installed, 0 to remove and 371 not upgraded.
Need to get 12.8 kB of archives.
After this operation, 79.9 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports bionic-updates/universe arm64 netfilter-
persistent all 1.0.4+nmu2ubuntu1 [6584 B]
Get:2 http://ports.ubuntu.com/ubuntu-ports bionic-updates/universe arm64 iptables-
persistent all 1.0.4+nmu2ubuntu1 [6240 B]
Fetched 12.8 kB in 0s (75.9 kB/s)
Preconfiguring packages ...
Selecting previously unselected package netfilter-persistent.
(Reading database ... 154029 files and directories currently installed.)
Preparing to unpack ../netfilter-persistent_1.0.4+nmu2ubuntu1_all.deb ...
Unpacking netfilter-persistent (1.0.4+nmu2ubuntu1) ...
Selecting previously unselected package iptables-persistent.
Preparing to unpack ../iptables-persistent_1.0.4+nmu2ubuntu1_all.deb ...
Unpacking iptables-persistent (1.0.4+nmu2ubuntu1) ...
Setting up netfilter-persistent (1.0.4+nmu2ubuntu1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/netfilter-
persistent.service → /lib/systemd/system/netfilter-persistent.service.
update-rc.d: warning: start and stop actions are no longer supported; falling back to
defaults
Setting up iptables-persistent (1.0.4+nmu2ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for systemd (237-3ubuntu10.25) ...

```

En aquesta part de l'output és pot veure part de l'execució del script `conf_network_master.sh`:

```

RTNETLINK answers: Cannot assign requested address
net.ipv4.ip_forward = 1

```

Al següent tros de codi es pot veure com s'instal·la el software sshpass i com es configura el servidor SSH:

```

Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:7.6p1-4ubuntu0.3).
The following packages were automatically installed and are no longer required:
  libdrm-freedreno1 libllvm6.0
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  sshpass
0 upgraded, 1 newly installed, 0 to remove and 371 not upgraded.
Need to get 10.3 kB of archives.
After this operation, 29.7 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 sshpass arm64 1.06-1
[10.3 kB]
Fetched 10.3 kB in 0s (91.4 kB/s)
Selecting previously unselected package sshpass.
(Reading database ... 154047 files and directories currently installed.)
Preparing to unpack ../sshpass_1.06-1_arm64.deb ...
Unpacking sshpass (1.06-1) ...
Setting up sshpass (1.06-1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Failed to enable unit: Refusing to operate on linked unit file sshd.service
# master:22 SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.3
# master:22 SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.3
# master:22 SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.3
Reading package lists... Done
Building dependency tree
Reading state information... Done

```

Aquest tros de l'output s'ha omès, per a que la memòria no ocupi un numero de pagines innecessàriament exagerat, hi ha un error que s'ha solucionat en quant s'ha trobat al dur a terme aquesta avaluació, s'ha pogut veure com la configuració del servidor VNC no es duia a terme, perquè no s'havia ficat correctament el PATH absolut cap al fitxer de configuració:

```
/////
```

```

CODI ESBORRAT ON APAREIXIA LA  DESINSTAL·LACIÓ DE SOFTWARE I LA INSTAL·LACIÓ DE
L'ENTORN GRÀFIC XFCE

```

```
/////
```

La instal·lació i configuració del servidor VNC es troba a continuació:

```

Processing triggers for initramfs-tools (0.130ubuntu3.8) ...
update-initramfs: Generating /boot/initrd.img-3.16.72-46

```



```

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 345 not upgraded.
Need to get 569 kB of archives.
After this operation, 1654 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 tightvncserver arm64
1.3.10-0ubuntu4 [569 kB]
Fetched 569 kB in 0s (1965 kB/s)
Selecting previously unselected package tightvncserver.
(Reading database ... 141016 files and directories currently installed.)
Preparing to unpack ../tightvncserver_1.3.10-0ubuntu4_arm64.deb ...
Unpacking tightvncserver (1.3.10-0ubuntu4) ...
Setting up tightvncserver (1.3.10-0ubuntu4) ...
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 (2.8.3-2ubuntu0.1) ...
./init_master.sh: line 192: $(eval echo "$name")/.vnc/xstartup: No such file or
directory

```

Al següent tros de output s'instal·la el servidor NFS, s'habilita el seu dimoni i s'exporta el directori */home*:

```

Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  keyutils libevent-2.1-6 libnfsidmap2 libtirpc1 nfs-common rpcbind
Suggested packages:
  open-iscsi watchdog
The following NEW packages will be installed:
  keyutils libevent-2.1-6 libnfsidmap2 libtirpc1 nfs-common nfs-kernel-server rpcbind
0 upgraded, 7 newly installed, 0 to remove and 345 not upgraded.
Need to get 552 kB of archives.
After this operation, 1970 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports bionic/main arm64 keyutils arm64 1.5.9-
9.2ubuntu2 [45.6 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports bionic/main arm64 libevent-2.1-6 arm64
2.1.8-stable-4build1 [109 kB]
Get:3 http://ports.ubuntu.com/ubuntu-ports bionic/main arm64 libnfsidmap2 arm64 0.25-
5.1 [24.5 kB]

```

```

Get:4 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 libtirpc1 arm64
0.2.5-1.2ubuntu0.1 [65.7 kB]
Get:5 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 rpcbind arm64
0.2.3-0.6ubuntu0.18.04.1 [37.9 kB]
Get:6 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 nfs-common arm64
1:1.3.4-2.1ubuntu5.3 [186 kB]
Get:7 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 nfs-kernel-server
arm64 1:1.3.4-2.1ubuntu5.3 [82.7 kB]
Fetched 552 kB in 0s (1196 kB/s)
Selecting previously unselected package keyutils.
(Reading database ... 141035 files and directories currently installed.)
Preparing to unpack .../0-keyutils_1.5.9-9.2ubuntu2_arm64.deb ...
Unpacking keyutils (1.5.9-9.2ubuntu2) ...
Selecting previously unselected package libevent-2.1-6:arm64.
Preparing to unpack .../1-libevent-2.1-6_2.1.8-stable-4build1_arm64.deb ...
Unpacking libevent-2.1-6:arm64 (2.1.8-stable-4build1) ...
Selecting previously unselected package libnfsidmap2:arm64.
Preparing to unpack .../2-libnfsidmap2_0.25-5.1_arm64.deb ...
Unpacking libnfsidmap2:arm64 (0.25-5.1) ...
Selecting previously unselected package libtirpc1:arm64.
Preparing to unpack .../3-libtirpc1_0.2.5-1.2ubuntu0.1_arm64.deb ...
Unpacking libtirpc1:arm64 (0.2.5-1.2ubuntu0.1) ...
Selecting previously unselected package rpcbind.
Preparing to unpack .../4-rpcbind_0.2.3-0.6ubuntu0.18.04.1_arm64.deb ...
Unpacking rpcbind (0.2.3-0.6ubuntu0.18.04.1) ...
Selecting previously unselected package nfs-common.
Preparing to unpack .../5-nfs-common_1%3a1.3.4-2.1ubuntu5.3_arm64.deb ...
Unpacking nfs-common (1:1.3.4-2.1ubuntu5.3) ...
Selecting previously unselected package nfs-kernel-server.
Preparing to unpack .../6-nfs-kernel-server_1%3a1.3.4-2.1ubuntu5.3_arm64.deb ...
Unpacking nfs-kernel-server (1:1.3.4-2.1ubuntu5.3) ...
Setting up libevent-2.1-6:arm64 (2.1.8-stable-4build1) ...
Setting up libnfsidmap2:arm64 (0.25-5.1) ...
Setting up keyutils (1.5.9-9.2ubuntu2) ...
Setting up libtirpc1:arm64 (0.2.5-1.2ubuntu0.1) ...
Setting up rpcbind (0.2.3-0.6ubuntu0.18.04.1) ...
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 (1:1.3.4-2.1ubuntu5.3) ...

Creating config file /etc/idmapd.conf with new version
Adding system user `statd' (UID 124) ...
Adding new user `statd' (UID 124) 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 (1:1.3.4-2.1ubuntu5.3) ...
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 (237-3ubuntu10.25) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
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

```

En aquesta part s'instal·la Munge, s'habilita el seu dimoni i es crea la clau que farà servir:

```

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 345 not upgraded.
Need to get 98.3 kB of archives.
After this operation, 361 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 libmunge2 arm64
0.5.13-1 [17.7 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 munge arm64 0.5.13-1
[80.6 kB]
Fetched 98.3 kB in 0s (357 kB/s)
Selecting previously unselected package libmunge2.
(Reading database ... 141204 files and directories currently installed.)
Preparing to unpack ../libmunge2_0.5.13-1_arm64.deb ...
Unpacking libmunge2 (0.5.13-1) ...
Selecting previously unselected package munge.
Preparing to unpack ../munge_0.5.13-1_arm64.deb ...
Unpacking munge (0.5.13-1) ...
Setting up libmunge2 (0.5.13-1) ...
Setting up munge (0.5.13-1) ...
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 (237-3ubuntu10.25) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
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, 0.000466056 s, 2.2 MB/s
Created symlink /etc/systemd/system/shutdown.target.wants/clean_tmp_hosts.service →
/etc/systemd/system/clean_tmp_hosts.service.

```

A la següent secció s'instal·la i configura Slurm, a banda, es comparteix el seu directori de configuració mitjançant NFS:

```

Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  cpp-7 freeipmi-common g++-7 gcc-7 gcc-7-base libaec0 libasan4 libdbi1 libfreeipmi16
  libgcc-7-dev libgfortran4 libhdf5-100 libhwloc-plugins libhwloc5
  libipmimonitoring5a liblua5.1-0 librrd8 libslurmdb32 libstdc++-7-dev libsz2 libubsan0
  ocl-icd-libopencl1 slurm-client slurm-wlm-basic-plugins slurmctld slurmd
Suggested packages:
  gcc-7-locales freeipmi-tools gcc-7-doc libstdc++6-7-dbg libgcc1-dbg libgomp1-dbg
  libitm1-dbg libatomic1-dbg libasan4-dbg liblsan0-dbg libtsan0-dbg libubsan0-dbg
  libcilkrt5-dbg libmpx2-dbg libquadmath0-dbg libhwloc-contrib-plugins libstdc++-7-
  doc opencl-icd
The following NEW packages will be installed:
  freeipmi-common libaec0 libdbi1 libfreeipmi16 libgfortran4 libhdf5-100 libhwloc-
  plugins libhwloc5 libipmimonitoring5a liblua5.1-0 librrd8 libslurmdb32 libsz2
  ocl-icd-libopencl1 slurm-client slurm-wlm slurm-wlm-basic-plugins slurmctld slurmd
The following packages will be upgraded:
  cpp-7 g++-7 gcc-7 gcc-7-base libasan4 libgcc-7-dev libstdc++-7-dev libubsan0
8 upgraded, 19 newly installed, 0 to remove and 337 not upgraded.
Need to get 31.5 MB of archives.
After this operation, 26.7 MB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 freeipmi-common
arm64 1.4.11-1.1ubuntu4.1 [174 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 libasan4 arm64
7.5.0-3ubuntu1~18.04 [335 kB]
Get:3 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 libubsan0 arm64
7.5.0-3ubuntu1~18.04 [117 kB]
Get:4 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 g++-7 arm64 7.5.0-
3ubuntu1~18.04 [8077 kB]
Get:5 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 gcc-7 arm64 7.5.0-
3ubuntu1~18.04 [7772 kB]
Get:6 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 libstdc++-7-dev
arm64 7.5.0-3ubuntu1~18.04 [1471 kB]

```

```

Get:7 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 libgcc-7-dev
arm64 7.5.0-3ubuntu1~18.04 [833 kB]
Get:8 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 cpp-7 arm64 7.5.0-
3ubuntu1~18.04 [7046 kB]
Get:9 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 gcc-7-base arm64
7.5.0-3ubuntu1~18.04 [18.3 kB]
Get:10 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 libaec0 arm64 0.3.2-
2 [16.7 kB]
Get:11 http://ports.ubuntu.com/ubuntu-ports bionic/main arm64 libdb11 arm64 0.9.0-5
[23.5 kB]
Get:12 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 libgfortran4
arm64 7.5.0-3ubuntu1~18.04 [248 kB]
Get:13 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 libsz2 arm64 0.3.2-
2 [4926 B]
Get:14 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 libhdf5-100 arm64
1.10.0-patch1+docs-4 [1010 kB]
Get:15 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 liblua5.1-0 arm64
5.1.5-8.1build2 [90.8 kB]
Get:16 http://ports.ubuntu.com/ubuntu-ports bionic/main arm64 librrd8 arm64 1.7.0-
1build1 [136 kB]
Get:17 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 libslurmdb32 arm64
17.11.2-1build1 [497 kB]
Get:18 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 libfreeipmi16
arm64 1.4.11-1.1ubuntu4.1 [729 kB]
Get:19 http://ports.ubuntu.com/ubuntu-ports bionic/main arm64 ocl-icd-libopencl1 arm64
2.2.11-1ubuntu1 [29.2 kB]
Get:20 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 libhwloc5 arm64
1.11.9-1 [84.4 kB]
Get:21 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 libhwloc-plugins
arm64 1.11.9-1 [11.2 kB]
Get:22 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64
libipmimonitoring5a arm64 1.4.11-1.1ubuntu4.1 [34.5 kB]
Get:23 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 slurm-wlm-basic-
plugins arm64 17.11.2-1build1 [1028 kB]
Get:24 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 slurm-client arm64
17.11.2-1build1 [817 kB]
Get:25 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 slurmd arm64 17.11.2-
1build1 [322 kB]
Get:26 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 slurmctld arm64
17.11.2-1build1 [526 kB]
Get:27 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 slurm-wlm arm64
17.11.2-1build1 [12.2 kB]
Fetched 31.5 MB in 2s (18.7 MB/s)
Selecting previously unselected package freeipmi-common.
(Reading database ... 141244 files and directories currently installed.)
Preparing to unpack .../00-freeipmi-common_1.4.11-1.1ubuntu4.1_arm64.deb ...
Unpacking freeipmi-common (1.4.11-1.1ubuntu4.1) ...
Preparing to unpack .../01-libasan4_7.5.0-3ubuntu1~18.04_arm64.deb ...
Unpacking libasan4:arm64 (7.5.0-3ubuntu1~18.04) over (7.4.0-1ubuntu1~18.04.1) ...
Preparing to unpack .../02-libubsan0_7.5.0-3ubuntu1~18.04_arm64.deb ...
Unpacking libubsan0:arm64 (7.5.0-3ubuntu1~18.04) over (7.4.0-1ubuntu1~18.04.1) ...
Preparing to unpack .../03-g++-7_7.5.0-3ubuntu1~18.04_arm64.deb ...
Unpacking g++-7 (7.5.0-3ubuntu1~18.04) over (7.4.0-1ubuntu1~18.04.1) ...
Preparing to unpack .../04-gcc-7_7.5.0-3ubuntu1~18.04_arm64.deb ...

```

```

Unpacking gcc-7 (7.5.0-3ubuntu1~18.04) over (7.4.0-1ubuntu1~18.04.1) ...
Preparing to unpack .../05-libstdc++-7-dev_7.5.0-3ubuntu1~18.04_arm64.deb ...
Unpacking libstdc++-7-dev:arm64 (7.5.0-3ubuntu1~18.04) over (7.4.0-1ubuntu1~18.04.1)
...
Preparing to unpack .../06-libgcc-7-dev_7.5.0-3ubuntu1~18.04_arm64.deb ...
Unpacking libgcc-7-dev:arm64 (7.5.0-3ubuntu1~18.04) over (7.4.0-1ubuntu1~18.04.1) ...
Preparing to unpack .../07-cpp-7_7.5.0-3ubuntu1~18.04_arm64.deb ...
Unpacking cpp-7 (7.5.0-3ubuntu1~18.04) over (7.4.0-1ubuntu1~18.04.1) ...
Preparing to unpack .../08-gcc-7-base_7.5.0-3ubuntu1~18.04_arm64.deb ...
Unpacking gcc-7-base:arm64 (7.5.0-3ubuntu1~18.04) over (7.4.0-1ubuntu1~18.04.1) ...
Selecting previously unselected package libaec0:arm64.
Preparing to unpack .../09-libaec0_0.3.2-2_arm64.deb ...
Unpacking libaec0:arm64 (0.3.2-2) ...
Selecting previously unselected package libdbil:arm64.
Preparing to unpack .../10-libdbil_0.9.0-5_arm64.deb ...
Unpacking libdbil:arm64 (0.9.0-5) ...
Selecting previously unselected package libgfortran4:arm64.
Preparing to unpack .../11-libgfortran4_7.5.0-3ubuntu1~18.04_arm64.deb ...
Unpacking libgfortran4:arm64 (7.5.0-3ubuntu1~18.04) ...
Selecting previously unselected package libsz2:arm64.
Preparing to unpack .../12-libsz2_0.3.2-2_arm64.deb ...
Unpacking libsz2:arm64 (0.3.2-2) ...
Selecting previously unselected package libhdf5-100:arm64.
Preparing to unpack .../13-libhdf5-100_1.10.0-patch1+docs-4_arm64.deb ...
Unpacking libhdf5-100:arm64 (1.10.0-patch1+docs-4) ...
Selecting previously unselected package liblua5.1-0:arm64.
Preparing to unpack .../14-liblua5.1-0_5.1.5-8.1build2_arm64.deb ...
Unpacking liblua5.1-0:arm64 (5.1.5-8.1build2) ...
Selecting previously unselected package librrd8:arm64.
Preparing to unpack .../15-librrd8_1.7.0-1build1_arm64.deb ...
Unpacking librrd8:arm64 (1.7.0-1build1) ...
Selecting previously unselected package libslurmdb32.
Preparing to unpack .../16-libslurmdb32_17.11.2-1build1_arm64.deb ...
Unpacking libslurmdb32 (17.11.2-1build1) ...
Selecting previously unselected package libfreeipmi16.
Preparing to unpack .../17-libfreeipmi16_1.4.11-1.1ubuntu4.1_arm64.deb ...
Unpacking libfreeipmi16 (1.4.11-1.1ubuntu4.1) ...
Selecting previously unselected package ocl-icd-libopencl1:arm64.
Preparing to unpack .../18-ocl-icd-libopencl1_2.2.11-1ubuntu1_arm64.deb ...
Unpacking ocl-icd-libopencl1:arm64 (2.2.11-1ubuntu1) ...
Selecting previously unselected package libhwloc5:arm64.
Preparing to unpack .../19-libhwloc5_1.11.9-1_arm64.deb ...
Unpacking libhwloc5:arm64 (1.11.9-1) ...
Selecting previously unselected package libhwloc-plugins.
Preparing to unpack .../20-libhwloc-plugins_1.11.9-1_arm64.deb ...
Unpacking libhwloc-plugins (1.11.9-1) ...

```

```

Selecting previously unselected package libipmmonitoring5a.
Preparing to unpack .../21-libipmmonitoring5a_1.4.11-1.1ubuntu4.1_arm64.deb ...
Unpacking libipmmonitoring5a (1.4.11-1.1ubuntu4.1) ...
Selecting previously unselected package slurm-wlm-basic-plugins.
Preparing to unpack .../22-slurm-wlm-basic-plugins_17.11.2-1build1_arm64.deb ...
Unpacking slurm-wlm-basic-plugins (17.11.2-1build1) ...
Selecting previously unselected package slurm-client.
Preparing to unpack .../23-slurm-client_17.11.2-1build1_arm64.deb ...
Unpacking slurm-client (17.11.2-1build1) ...
Selecting previously unselected package slurmd.
Preparing to unpack .../24-slurmd_17.11.2-1build1_arm64.deb ...
Unpacking slurmd (17.11.2-1build1) ...
Selecting previously unselected package slurmctld.
Preparing to unpack .../25-slurmctld_17.11.2-1build1_arm64.deb ...
Unpacking slurmctld (17.11.2-1build1) ...
Selecting previously unselected package slurm-wlm.
Preparing to unpack .../26-slurm-wlm_17.11.2-1build1_arm64.deb ...
Unpacking slurm-wlm (17.11.2-1build1) ...
Setting up libaec0:arm64 (0.3.2-2) ...
Setting up libdbil:arm64 (0.9.0-5) ...
Setting up libhwloc5:arm64 (1.11.9-1) ...
Setting up librrd8:arm64 (1.7.0-1build1) ...
Setting up gcc-7-base:arm64 (7.5.0-3ubuntu1~18.04) ...
Setting up liblua5.1-0:arm64 (5.1.5-8.1build2) ...
Setting up freeipmi-common (1.4.11-1.1ubuntu4.1) ...
Setting up libfreeipmi16 (1.4.11-1.1ubuntu4.1) ...
Setting up ocl-icd-libopencl1:arm64 (2.2.11-1ubuntu1) ...
Setting up libslurmdb32 (17.11.2-1build1) ...
Setting up libsz2:arm64 (0.3.2-2) ...
Setting up libasan4:arm64 (7.5.0-3ubuntu1~18.04) ...
Setting up libgfortran4:arm64 (7.5.0-3ubuntu1~18.04) ...
Setting up libubsan0:arm64 (7.5.0-3ubuntu1~18.04) ...
Setting up libgcc-7-dev:arm64 (7.5.0-3ubuntu1~18.04) ...
Setting up cpp-7 (7.5.0-3ubuntu1~18.04) ...
Setting up libstdc++-7-dev:arm64 (7.5.0-3ubuntu1~18.04) ...
Setting up libipmmonitoring5a (1.4.11-1.1ubuntu4.1) ...
Setting up libhwloc-plugins (1.11.9-1) ...
Setting up libhdf5-100:arm64 (1.10.0-patch1+docs-4) ...
Setting up gcc-7 (7.5.0-3ubuntu1~18.04) ...
Setting up g++-7 (7.5.0-3ubuntu1~18.04) ...
Setting up slurm-wlm-basic-plugins (17.11.2-1build1) ...
Setting up slurmd (17.11.2-1build1) ...
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 (17.11.2-1build1) ...
Setting up slurmctld (17.11.2-1build1) ...
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 (17.11.2-1build1) ...
Processing triggers for systemd (237-3ubuntu10.25) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Synchronizing state of slurmctld.service with SysV service script with
/lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable slurmctld

```

A la següent secció s'instal·la MPICH:

```

Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  gfortran gfortran-7 hwloc-nox libgfortran-7-dev libmpich-dev libmpich12
Suggested packages:
  gfortran-doc gfortran-7-doc libgfortran4-dbg libcoarrays-dev blcr-util mpich-doc
The following NEW packages will be installed:
  gfortran gfortran-7 hwloc-nox libgfortran-7-dev libmpich-dev libmpich12 mpich
0 upgraded, 7 newly installed, 0 to remove and 337 not upgraded.
Need to get 10.0 MB of archives.
After this operation, 34.7 MB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 libgfortran-7-dev
arm64 7.5.0-3ubuntu1~18.04 [300 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 gfortran-7 arm64
7.5.0-3ubuntu1~18.04 [7465 kB]
Get:3 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 gfortran arm64
4:7.4.0-1ubuntu2.3 [1344 B]
Get:4 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 hwloc-nox arm64
1.11.9-1 [143 kB]
Get:5 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 libmpich12 arm64
3.3~a2-4 [740 kB]
Get:6 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 libmpich-dev arm64
3.3~a2-4 [1196 kB]
Get:7 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 mpich arm64 3.3~a2-4
[160 kB]
Fetched 10.0 MB in 1s (13.9 MB/s)
Selecting previously unselected package libgfortran-7-dev:arm64.
(Reading database ... 141608 files and directories currently installed.)
Preparing to unpack .../0-libgfortran-7-dev_7.5.0-3ubuntu1~18.04_arm64.deb ...
Unpacking libgfortran-7-dev:arm64 (7.5.0-3ubuntu1~18.04) ...
Selecting previously unselected package gfortran-7.
Preparing to unpack .../1-gfortran-7_7.5.0-3ubuntu1~18.04_arm64.deb ...
Unpacking gfortran-7 (7.5.0-3ubuntu1~18.04) ...

```



```

Selecting previously unselected package gfortran.
Preparing to unpack .../2-gfortran_4%3a7.4.0-1ubuntu2.3_arm64.deb ...
Unpacking gfortran (4:7.4.0-1ubuntu2.3) ...
Selecting previously unselected package hwloc-nox.
Preparing to unpack .../3-hwloc-nox_1.11.9-1_arm64.deb ...
Unpacking hwloc-nox (1.11.9-1) ...
Selecting previously unselected package libmpich12:arm64.
Preparing to unpack .../4-libmpich12_3.3~a2-4_arm64.deb ...
Unpacking libmpich12:arm64 (3.3~a2-4) ...
Selecting previously unselected package libmpich-dev.
Preparing to unpack .../5-libmpich-dev_3.3~a2-4_arm64.deb ...
Unpacking libmpich-dev (3.3~a2-4) ...
Selecting previously unselected package mpich.
Preparing to unpack .../6-mpich_3.3~a2-4_arm64.deb ...
Unpacking mpich (3.3~a2-4) ...
Setting up hwloc-nox (1.11.9-1) ...
Setting up libmpich12:arm64 (3.3~a2-4) ...
Setting up libgfortran-7-dev:arm64 (7.5.0-3ubuntu1~18.04) ...
Setting up gfortran-7 (7.5.0-3ubuntu1~18.04) ...
Setting up gfortran (4:7.4.0-1ubuntu2.3) ...
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 mpich (3.3~a2-4) ...
update-alternatives: using /usr/bin/mpirun.mpich to provide /usr/bin/mpirun (mpirun)
in auto mode
Setting up libmpich-dev (3.3~a2-4) ...
update-alternatives: using /usr/include/mpich to provide /usr/include/mpi (mpi) in auto
mode
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
RTNETLINK answers: File exists

```

Amb el següent output podem validar la correcta instal·lació i configuració del servidor dnsmasq:

```

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 337 not upgraded.
Need to get 16.2 kB of archives.
After this operation, 73.7 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 dnsmasq all 2.79-1
[16.2 kB]

```

```

Fetched 16.2 kB in 0s (115 kB/s)
Selecting previously unselected package dnsmasq.
(Reading database ... 141762 files and directories currently installed.)
Preparing to unpack .../dnsmasq_2.79-1_all.deb ...
Unpacking dnsmasq (2.79-1) ...
Setting up dnsmasq (2.79-1) ...
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 Fri 2020-11-06 03:54:09 CET; 25ms ago
   Process: 13472 ExecStart=/etc/init.d/dnsmasq systemd-exec (code=exited, status=2)
   Process: 13470 ExecStartPre=/usr/sbin/dnsmasq --test (code=exited, status=0/SUCCESS)

Nov 06 03:54:09 master systemd[1]: Starting dnsmasq - A lightweight DHCP and caching
DNS server...
Nov 06 03:54:09 master dnsmasq[13470]: dnsmasq: syntax check OK.
Nov 06 03:54:09 master dnsmasq[13472]: dnsmasq: failed to create listening socket for
port 53: Address already in use
Nov 06 03:54:09 master dnsmasq[13472]: failed to create listening socket for port 53:
Address already in use
Nov 06 03:54:09 master dnsmasq[13472]: FAILED to start up
Nov 06 03:54:09 master systemd[1]: dnsmasq.service: Control process exited, code=exited
status=2
Nov 06 03:54:09 master systemd[1]: dnsmasq.service: Failed with result 'exit-code'.
Nov 06 03:54:09 master systemd[1]: Failed to start dnsmasq - A lightweight DHCP and
caching DNS server.
Processing triggers for systemd (237-3ubuntu10.25) ...
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/systemd-resolved.service.
Removed /etc/systemd/system/dbus-org.freedesktop.resolve1.service.
cp: missing destination file operand after '/etc/dnsmasq.conf'
Try 'cp --help' for more information.
Synchronizing state of dnsmasq.service with SysV service script with
/lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable dnsmasq

```

En aquesta part s'instal·la el software de monitoratge, però com ja s'ha dit, sha tret per escurçar el numero de pàgines del document.

//////////

CODI ESBORRAT ON APAREIXIA LA INSTAL·LACIÓ DEL SOFTWARE DE MONITORATGE

```
//////////
```

A partir d'aquí fins al final, s'instal·la i configura l'idioma del sistema operatiu i els programes, es pot veure com a les últimes línies passa de parlar en anglès a castellà:

```
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:
  language-pack-es language-pack-es-base language-pack-gnome-es language-pack-gnome-
es-base wspanish
  0 actualizados, 5 nuevos se instalarán, 0 para eliminar y 329 no actualizados.
Se necesita descargar 8.676 kB de archivos.
Se utilizarán 35,9 MB de espacio de disco adicional después de esta operación.
Des:1 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 language-pack-es-
base all 1:18.04+20180712 [2.845 kB]
Des:2 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 language-pack-es
all 1:18.04+20200702 [2.002 kB]
Des:3 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 language-pack-
gnome-es-base all 1:18.04+20180712 [2.879 kB]
Des:4 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 language-pack-
gnome-es all 1:18.04+20200702 [739 kB]
Des:5 http://ports.ubuntu.com/ubuntu-ports bionic/main arm64 wspanish all 1.0.27 [211
kB]
Descargados 8.676 kB en 1s (8.846 kB/s)
Preconfigurando paquetes ...
Seleccionando el paquete language-pack-es-base previamente no seleccionado.
(Leyendo la base de datos ... 157928 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar ../language-pack-es-base_1%3a18.04+20180712_all.deb ...
Desempaquetando language-pack-es-base (1:18.04+20180712) ...
Seleccionando el paquete language-pack-es previamente no seleccionado.
Preparando para desempaquetar ../language-pack-es_1%3a18.04+20200702_all.deb ...
Desempaquetando language-pack-es (1:18.04+20200702) ...
Reemplazando ficheros del paquete antiguo language-pack-es-base (1:18.04+20180712) ...
Seleccionando el paquete language-pack-gnome-es-base previamente no seleccionado.
Preparando para desempaquetar ../language-pack-gnome-es-
base_1%3a18.04+20180712_all.deb ...
Desempaquetando language-pack-gnome-es-base (1:18.04+20180712) ...
Seleccionando el paquete language-pack-gnome-es previamente no seleccionado.
Preparando para desempaquetar ../language-pack-gnome-es_1%3a18.04+20200702_all.deb
...
Desempaquetando language-pack-gnome-es (1:18.04+20200702) ...
Reemplazando ficheros del paquete antiguo language-pack-gnome-es-base
(1:18.04+20180712) ...
Seleccionando el paquete wspanish previamente no seleccionado.
Preparando para desempaquetar ../wspanish_1.0.27_all.deb ...
```

```

Desempaquetando wspanish (1.0.27) ...
Configurando wspanish (1.0.27) ...
Configurando language-pack-es (1:18.04+20200702) ...
Configurando language-pack-es-base (1:18.04+20180712) ...
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.
Configurando language-pack-gnome-es (1:18.04+20200702) ...
Configurando language-pack-gnome-es-base (1:18.04+20180712) ...
Procesando disparadores para cracklib-runtime (2.9.2-5build1) ...
Procesando disparadores para man-db (2.8.3-2ubuntu0.1) ...
Procesando disparadores para dictionaries-common (1.27.2) ...
Procesando disparadores para bamfdaemon (0.5.3+18.04.20180207.2-0ubuntu1) ...
Rebuilding /usr/share/applications/bamf-2.index...

```

5.2 Node slave

Per començar és mostrarà l'output del fitxer *add_slave.sh*, i a continuació fins a acabar l'output del script *init_slave.sh*.

De tot el que fa el script *add_slave.sh*, només es pot apreciar com s'afegeix la clau al slave *odroid1*:

```

# odroid1:22 SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.3
# odroid1:22 SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.3
# odroid1:22 SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.3

```

```

/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/odroid/.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 'odroid@odroid1'"
and check to make sure that only the key(s) you wanted were added.

A partir d'aquí, es passarà a avaluar la execució de `init_slave.sh`, al igual que ha `init_master.sh`, s'ha tret la part de les desinstal·lacions.

En aquesta primera secció podem veure com es deshabilita `unattended-upgrades` i com es desinstal·la:

```

Pseudo-terminal will not be allocated because stdin is not a terminal.

[sudo] password for odroid: Synchronizing state of unattended-upgrades.service with
SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable unattended-upgrades

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Reading package lists...
Building dependency tree...
Reading state information...
The following packages were automatically installed and are no longer required:
  libdrm-freedreno1 libllvm6.0
Use 'sudo apt autoremove' to remove them.
The following packages will be REMOVED:
  unattended-upgrades
0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.
After this operation, 393 kB disk space will be freed.
(Reading database ... 153565 files and directories currently installed.)
Removing unattended-upgrades (1.1ubuntu1.18.04.11) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...

```

En aquest tros d'output, es configura la IP del *master* com a servidor DNS, es configuren les interfícies de xarxa i com es crida al servidor DHCP per modificar el *hostname* de forma automàtica:

```

Internet Systems Consortium DHCP Client 4.3.5
Copyright 2004-2016 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Invalid number of 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=0x68209a28)
DHCPREQUEST of 172.16.0.41 on eth0 to 255.255.255.255 port 67 (xid=0x289a2068)
DHCPOFFER of 172.16.0.41 from 172.16.0.1
DHCPACK of 172.16.0.41 from 172.16.0.1
bound to 172.16.0.41 -- renewal in 17148 seconds.

```

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

En aquesta part s'instal·la el dimoni *chrony*, a tal com ha passat al *master*, surt un error, però aquest tampoc afecta al seu funcionament:

```

Reading package lists...
Building dependency tree...
Reading state information...
The following packages were automatically installed and are no longer required:
  libdrm-freedreno1 libllvm6.0
Use 'sudo apt autoremove' to remove them.
Suggested packages:
  dnstools
The following NEW packages will be installed:
  chrony
0 upgraded, 1 newly installed, 0 to remove and 370 not upgraded.
Need to get 184 kB of archives.
After this operation, 464 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 chrony arm64 3.2-4ubuntu4.5 [184 kB]
debconf: unable to initialize frontend: Dialog
debconf: (Dialog frontend will not work on a dumb terminal, an emacs shell buffer, or
without a controlling terminal.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (This frontend requires a controlling tty.)
debconf: falling back to frontend: Teletype
dpkg-preconfigure: unable to re-open stdin:
Fetched 184 kB in 0s (528 kB/s)
Selecting previously unselected package chrony.
(Reading database ... 153542 files and directories currently installed.)
Preparing to unpack ../chrony_3.2-4ubuntu4.5_arm64.deb ...
Unpacking chrony (3.2-4ubuntu4.5) ...
Setting up chrony (3.2-4ubuntu4.5) ...
Creating '_chrony' system user/group for the chronyd daemon...
debconf: unable to initialize frontend: Dialog
debconf: (Dialog frontend will not work on a dumb terminal, an emacs shell buffer, or
without a controlling terminal.)

```

```

debconf: falling back to frontend: Readline

Creating config file /etc/chrony/chrony.conf with new version
debconf: unable to initialize frontend: Dialog
debconf: (Dialog frontend will not work on a dumb terminal, an emacs shell buffer, or
without a controlling terminal.)
debconf: falling back to frontend: Readline

Creating config file /etc/chrony/chrony.keys with new version
Created symlink /etc/systemd/system/chronyd.service →
/lib/systemd/system/chrony.service.
Created symlink /etc/systemd/system/multi-user.target.wants/chrony.service →
/lib/systemd/system/chrony.service.

Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for systemd (237-3ubuntu10.25) ...
Failed to enable unit: Refusing to operate on linked unit file chronyd.service

```

En aquesta part es soluciona el problema amb la clau faltant per fer l'actualització del sistema:

```

Warning: apt-key output should not be parsed (stdout is not a terminal)
Executing: /tmp/apt-key-gpghome.ZxRBduuqdv/gpg.1.sh -v --keyserver
keyserver.ubuntu.com --recv-keys 5360FB9DAB19BAC9
gpg: no running Dirmngr - starting '/usr/bin/dirmngr'
gpg: waiting for the dirmngr to come up ... (5s)
gpg: waiting for the dirmngr to come up ... (4s)
gpg: connection to dirmngr established
gpg: data source: http://162.213.33.9:11371
gpg: pub rsa4096/4047B34823AE12F8 2015-01-07 Mauro Ribeiro (mdrjr)
<mauro.ribeiro@hardkernel.com>
gpg: key 4047B34823AE12F8: "Mauro Ribeiro (mdrjr) <mauro.ribeiro@hardkernel.com>" 1 new
signature
gpg: no running gpg-agent - starting '/usr/bin/gpg-agent'
gpg: waiting for the agent to come up ... (5s)
gpg: waiting for the agent to come up ... (4s)
gpg: connection to agent established
gpg: Total number processed: 1
gpg: new signatures: 1
Hit:1 http://ports.ubuntu.com/ubuntu-ports bionic InRelease
Hit:2 http://archive.canonical.com/ubuntu bionic InRelease
Hit:3 http://ports.ubuntu.com/ubuntu-ports bionic-updates InRelease
Hit:4 http://ports.ubuntu.com/ubuntu-ports bionic-backports InRelease
Hit:5 http://ports.ubuntu.com/ubuntu-ports bionic-security InRelease
Get:6 http://deb.odroid.in/c2 bionic InRelease [2406 B]
Get:7 http://deb.odroid.in/c2 bionic/main arm64 Packages [16.6 kB]
Fetched 16.6 kB in 2s (7383 B/s)

```

Veient aquesta secció es pot validar la correcta instal·lació del software sysstat:

```
Reading package lists...
```

```

Reading package lists...
Building dependency tree...
Reading state information...
The following packages were automatically installed and are no longer required:
  libdrm-freedreno1 libllvm6.0
Use 'sudo apt autoremove' to remove them.
Suggested packages:
  isag
The following NEW packages will be installed:
  sysstat
0 upgraded, 1 newly installed, 0 to remove and 371 not upgraded.
Need to get 257 kB of archives.
After this operation, 1077 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 sysstat arm64
11.6.1-1ubuntu0.1 [257 kB]
debconf: unable to initialize frontend: Dialog
debconf: (Dialog frontend will not work on a dumb terminal, an emacs shell buffer, or
without a controlling terminal.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (This frontend requires a controlling tty.)
debconf: falling back to frontend: Teletype
dpkg-preconfigure: unable to re-open stdin:
Fetched 257 kB in 0s (615 kB/s)
Selecting previously unselected package sysstat.
(Reading database ... 153581 files and directories currently installed.)
Preparing to unpack .../sysstat_11.6.1-1ubuntu0.1_arm64.deb ...
Unpacking sysstat (11.6.1-1ubuntu0.1) ...
Setting up sysstat (11.6.1-1ubuntu0.1) ...
debconf: unable to initialize frontend: Dialog
debconf: (Dialog frontend will not work on a dumb terminal, an emacs shell buffer, or
without a controlling terminal.)
debconf: falling back to frontend: Readline

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 man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for systemd (237-3ubuntu10.25) ...
Reading package lists...
Building dependency tree...
Reading state information...
/////

AQUÍ ES TROBAVA EL CODI DE LES DESINSTALACIONS

```



```
/////
```

Es pot validar la instal·lació del client NFS, veien el següent tros de output:

```
Reading package lists...
Building dependency tree...
Reading state information...
Reading package lists...
Building dependency tree...
Reading state information...
The following additional packages will be installed:
  keyutils libevent-2.1-6 libnfsidmap2 libtirpc1 rpcbind
Suggested packages:
  open-iscsi watchdog
The following NEW packages will be installed:
  keyutils libevent-2.1-6 libnfsidmap2 libtirpc1 nfs-common rpcbind
0 upgraded, 6 newly installed, 0 to remove and 348 not upgraded.
Need to get 469 kB of archives.
After this operation, 1654 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports bionic/main arm64 keyutils arm64 1.5.9-9.2ubuntu2 [45.6 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports bionic/main arm64 libevent-2.1-6 arm64 2.1.8-stable-4build1 [109 kB]
Get:3 http://ports.ubuntu.com/ubuntu-ports bionic/main arm64 libnfsidmap2 arm64 0.25-5.1 [24.5 kB]
Get:4 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 libtirpc1 arm64 0.2.5-1.2ubuntu0.1 [65.7 kB]
Get:5 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 rpcbind arm64 0.2.3-0.6ubuntu0.18.04.1 [37.9 kB]
Get:6 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 nfs-common arm64 1:1.3.4-2.1ubuntu5.3 [186 kB]
debconf: unable to initialize frontend: Dialog
debconf: (Dialog frontend will not work on a dumb terminal, an emacs shell buffer, or without a controlling terminal.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (This frontend requires a controlling tty.)
debconf: falling back to frontend: Teletype
dpkg-preconfigure: unable to re-open stdin:
Fetched 469 kB in 0s (1351 kB/s)
Selecting previously unselected package keyutils.
(Reading database ... 148633 files and directories currently installed.)
Preparing to unpack .../0-keyutils_1.5.9-9.2ubuntu2_arm64.deb ...
Unpacking keyutils (1.5.9-9.2ubuntu2) ...
Selecting previously unselected package libevent-2.1-6:arm64.
Preparing to unpack .../1-libevent-2.1-6_2.1.8-stable-4build1_arm64.deb ...
Unpacking libevent-2.1-6:arm64 (2.1.8-stable-4build1) ...
Selecting previously unselected package libnfsidmap2:arm64.
Preparing to unpack .../2-libnfsidmap2_0.25-5.1_arm64.deb ...
```

```

Unpacking libnfsidmap2:arm64 (0.25-5.1) ...
Selecting previously unselected package libtirpc1:arm64.
Preparing to unpack .../3-libtirpc1_0.2.5-1.2ubuntu0.1_arm64.deb ...
Unpacking libtirpc1:arm64 (0.2.5-1.2ubuntu0.1) ...
Selecting previously unselected package rpcbind.
Preparing to unpack .../4-rpcbind_0.2.3-0.6ubuntu0.18.04.1_arm64.deb ...
Unpacking rpcbind (0.2.3-0.6ubuntu0.18.04.1) ...
Selecting previously unselected package nfs-common.
Preparing to unpack .../5-nfs-common_1%3a1.3.4-2.1ubuntu5.3_arm64.deb ...
Unpacking nfs-common (1:1.3.4-2.1ubuntu5.3) ...
Setting up libevent-2.1-6:arm64 (2.1.8-stable-4build1) ...
Setting up libnfsidmap2:arm64 (0.25-5.1) ...
Setting up keyutils (1.5.9-9.2ubuntu2) ...
Setting up libtirpc1:arm64 (0.2.5-1.2ubuntu0.1) ...
Setting up rpcbind (0.2.3-0.6ubuntu0.18.04.1) ...
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 (1:1.3.4-2.1ubuntu5.3) ...
debconf: unable to initialize frontend: Dialog
debconf: (Dialog frontend will not work on a dumb terminal, an emacs shell buffer, or
without a controlling terminal.)
debconf: falling back to frontend: Readline

Creating config file /etc/idmapd.conf with new version
debconf: unable to initialize frontend: Dialog
debconf: (Dialog frontend will not work on a dumb terminal, an emacs shell buffer, or
without a controlling terminal.)
debconf: falling back to frontend: Readline
Adding system user `statd' (UID 123) ...
Adding new user `statd' (UID 123) 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 (237-3ubuntu10.25) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...

```

A les següents línies es pot veure com s'instal·la *Munge*, i es copia la clau compartida pel servidor, reiniciant després el dimoni:

```

Reading package lists...
Building dependency tree...
Reading state information...

```

```

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 348 not upgraded.
Need to get 98.3 kB of archives.
After this operation, 361 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 libmunge2 arm64
0.5.13-1 [17.7 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 munge arm64 0.5.13-1
[80.6 kB]
debconf: unable to initialize frontend: Dialog
debconf: (Dialog frontend will not work on a dumb terminal, an emacs shell buffer, or
without a controlling terminal.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (This frontend requires a controlling tty.)
debconf: falling back to frontend: Teletype
dpkg-preconfigure: unable to re-open stdin:
Fetched 98.3 kB in 0s (321 kB/s)
Selecting previously unselected package libmunge2.
(Reading database ... 148770 files and directories currently installed.)
Preparing to unpack ../libmunge2_0.5.13-1_arm64.deb ...
Unpacking libmunge2 (0.5.13-1) ...
Selecting previously unselected package munge.
Preparing to unpack ../munge_0.5.13-1_arm64.deb ...
Unpacking munge (0.5.13-1) ...
Setting up libmunge2 (0.5.13-1) ...
Setting up munge (0.5.13-1) ...
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 (237-3ubuntu10.25) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
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, 0.00246703 s, 415 kB/s

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

```

Veient les següents línies podem observar com s'instal·la *Slurm* al *slave*:

```

Reading package lists...
Building dependency tree...

```

Reading state information...

The following additional packages will be installed:

```
cpp-7 freeipmi-common g++-7 gcc-7 gcc-7-base libaec0 libasan4 libdbil1
libfreeipmi16 libgcc-7-dev libgfortran4 libhdf5-100 libhwloc-plugins
libhwloc5 libipmimonitoring5a liblua5.1-0 librrd8 libslurmdb32
libstdc++-7-dev libs2 libubsan0 ocl-icd-libopencl1 slurm-client
slurm-wlm-basic-plugins slurmctld slurmd
```

Suggested packages:

```
gcc-7-locales freeipmi-tools gcc-7-doc libstdc++6-7-dbg libgcc1-dbg
libgomp1-dbg libitm1-dbg libatomic1-dbg libasan4-dbg liblsan0-dbg
libtsan0-dbg libubsan0-dbg libcilkrts5-dbg libmpx2-dbg libquadmath0-dbg
libhwloc-contrib-plugins libstdc++-7-doc opencl-icd
```

The following NEW packages will be installed:

```
freeipmi-common libaec0 libdbil1 libfreeipmi16 libgfortran4 libhdf5-100
libhwloc-plugins libhwloc5 libipmimonitoring5a liblua5.1-0 librrd8
libslurmdb32 libs2 ocl-icd-libopencl1 slurm-client slurm-wlm
slurm-wlm-basic-plugins slurmctld slurmd
```

The following packages will be upgraded:

```
cpp-7 g++-7 gcc-7 gcc-7-base libasan4 libgcc-7-dev libstdc++-7-dev libubsan0
```

8 upgraded, 19 newly installed, 0 to remove and 340 not upgraded.

Need to get 31.5 MB of archives.

After this operation, 26.7 MB of additional disk space will be used.

```
Get:1 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 freeipmi-common
arm64 1.4.11-1.lubuntu4.1 [174 kB]
```

```
Get:2 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 libasan4 arm64
7.5.0-3ubuntu1~18.04 [335 kB]
```

```
Get:3 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 libubsan0 arm64
7.5.0-3ubuntu1~18.04 [117 kB]
```

```
Get:4 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 g++-7 arm64 7.5.0-
3ubuntu1~18.04 [8077 kB]
```

```
Get:5 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 gcc-7 arm64 7.5.0-
3ubuntu1~18.04 [7772 kB]
```

```
Get:6 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 libstdc++-7-dev
arm64 7.5.0-3ubuntu1~18.04 [1471 kB]
```

```
Get:7 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 libgcc-7-dev
arm64 7.5.0-3ubuntu1~18.04 [833 kB]
```

```
Get:8 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 cpp-7 arm64 7.5.0-
3ubuntu1~18.04 [7046 kB]
```

```
Get:9 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 gcc-7-base arm64
7.5.0-3ubuntu1~18.04 [18.3 kB]
```

```
Get:10 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 libaec0 arm64 0.3.2-
2 [16.7 kB]
```

```
Get:11 http://ports.ubuntu.com/ubuntu-ports bionic/main arm64 libdbil1 arm64 0.9.0-5
[23.5 kB]
```

```
Get:12 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 libgfortran4
arm64 7.5.0-3ubuntu1~18.04 [248 kB]
```

```
Get:13 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 libs2 arm64 0.3.2-
2 [4926 B]
```

```
Get:14 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 libhdf5-100 arm64
1.10.0-patch1+docs-4 [1010 kB]
```

```
Get:15 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 liblua5.1-0 arm64
5.1.5-8.1build2 [90.8 kB]
```

```

Get:16 http://ports.ubuntu.com/ubuntu-ports bionic/main arm64 librrd8 arm64 1.7.0-
1build1 [136 kB]
Get:17 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 libslurmdb32 arm64
17.11.2-1build1 [497 kB]
Get:18 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 libfreeipmi16
arm64 1.4.11-1.1ubuntu4.1 [729 kB]
Get:19 http://ports.ubuntu.com/ubuntu-ports bionic/main arm64 ocl-icd-libopencl1 arm64
2.2.11-1ubuntu1 [29.2 kB]
Get:20 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 libhwloc5 arm64
1.11.9-1 [84.4 kB]
Get:21 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 libhwloc-plugins
arm64 1.11.9-1 [11.2 kB]
Get:22 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64
libipmimonitoring5a arm64 1.4.11-1.1ubuntu4.1 [34.5 kB]
Get:23 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 slurm-wlm-basic-
plugins arm64 17.11.2-1build1 [1028 kB]
Get:24 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 slurm-client arm64
17.11.2-1build1 [817 kB]
Get:25 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 slurmd arm64 17.11.2-
1build1 [322 kB]
Get:26 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 slurmctld arm64
17.11.2-1build1 [526 kB]
Get:27 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 slurm-wlm arm64
17.11.2-1build1 [12.2 kB]
debconf: unable to initialize frontend: Dialog
debconf: (Dialog frontend will not work on a dumb terminal, an emacs shell buffer, or
without a controlling terminal.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (This frontend requires a controlling tty.)
debconf: falling back to frontend: Teletype
dpkg-preconfigure: unable to re-open stdin:
Fetched 31.5 MB in 3s (9870 kB/s)
Selecting previously unselected package freeipmi-common.
(Reading database ... 148810 files and directories currently installed.)
Preparing to unpack .../00-freeipmi-common_1.4.11-1.1ubuntu4.1_arm64.deb ...
Unpacking freeipmi-common (1.4.11-1.1ubuntu4.1) ...
Preparing to unpack .../01-libasan4_7.5.0-3ubuntu1~18.04_arm64.deb ...
Unpacking libasan4:arm64 (7.5.0-3ubuntu1~18.04) over (7.4.0-1ubuntu1~18.04.1) ...
Preparing to unpack .../02-libubsan0_7.5.0-3ubuntu1~18.04_arm64.deb ...
Unpacking libubsan0:arm64 (7.5.0-3ubuntu1~18.04) over (7.4.0-1ubuntu1~18.04.1) ...
Preparing to unpack .../03-g++-7_7.5.0-3ubuntu1~18.04_arm64.deb ...
Unpacking g++-7 (7.5.0-3ubuntu1~18.04) over (7.4.0-1ubuntu1~18.04.1) ...
Preparing to unpack .../04-gcc-7_7.5.0-3ubuntu1~18.04_arm64.deb ...
Unpacking gcc-7 (7.5.0-3ubuntu1~18.04) over (7.4.0-1ubuntu1~18.04.1) ...
Preparing to unpack .../05-libstdc++-7-dev_7.5.0-3ubuntu1~18.04_arm64.deb ...
Unpacking libstdc++-7-dev:arm64 (7.5.0-3ubuntu1~18.04) over (7.4.0-1ubuntu1~18.04.1)
...
Preparing to unpack .../06-libgcc-7-dev_7.5.0-3ubuntu1~18.04_arm64.deb ...
Unpacking libgcc-7-dev:arm64 (7.5.0-3ubuntu1~18.04) over (7.4.0-1ubuntu1~18.04.1) ...
Preparing to unpack .../07-cpp-7_7.5.0-3ubuntu1~18.04_arm64.deb ...

```

```
Unpacking cpp-7 (7.5.0-3ubuntu1~18.04) over (7.4.0-1ubuntu1~18.04.1) ...
Preparing to unpack .../08-gcc-7-base_7.5.0-3ubuntu1~18.04_arm64.deb ...
Unpacking gcc-7-base:arm64 (7.5.0-3ubuntu1~18.04) over (7.4.0-1ubuntu1~18.04.1) ...
Selecting previously unselected package libaec0:arm64.
Preparing to unpack .../09-libaec0_0.3.2-2_arm64.deb ...
Unpacking libaec0:arm64 (0.3.2-2) ...
Selecting previously unselected package libdbil:arm64.
Preparing to unpack .../10-libdbil_0.9.0-5_arm64.deb ...
Unpacking libdbil:arm64 (0.9.0-5) ...
Selecting previously unselected package libgfortran4:arm64.
Preparing to unpack .../11-libgfortran4_7.5.0-3ubuntu1~18.04_arm64.deb ...
Unpacking libgfortran4:arm64 (7.5.0-3ubuntu1~18.04) ...
Selecting previously unselected package libsz2:arm64.
Preparing to unpack .../12-libsz2_0.3.2-2_arm64.deb ...
Unpacking libsz2:arm64 (0.3.2-2) ...
Selecting previously unselected package libhdf5-100:arm64.
Preparing to unpack .../13-libhdf5-100_1.10.0-patch1+docs-4_arm64.deb ...
Unpacking libhdf5-100:arm64 (1.10.0-patch1+docs-4) ...
Selecting previously unselected package liblua5.1-0:arm64.
Preparing to unpack .../14-liblua5.1-0_5.1.5-8.1build2_arm64.deb ...
Unpacking liblua5.1-0:arm64 (5.1.5-8.1build2) ...
Selecting previously unselected package librrd8:arm64.
Preparing to unpack .../15-librrd8_1.7.0-1build1_arm64.deb ...
Unpacking librrd8:arm64 (1.7.0-1build1) ...
Selecting previously unselected package libslurmdb32.
Preparing to unpack .../16-libslurmdb32_17.11.2-1build1_arm64.deb ...
Unpacking libslurmdb32 (17.11.2-1build1) ...
Selecting previously unselected package libfreeipmi16.
Preparing to unpack .../17-libfreeipmi16_1.4.11-1.1ubuntu4.1_arm64.deb ...
Unpacking libfreeipmi16 (1.4.11-1.1ubuntu4.1) ...
Selecting previously unselected package ocl-icd-libopencl1:arm64.
Preparing to unpack .../18-ocl-icd-libopencl1_2.2.11-1ubuntu1_arm64.deb ...
Unpacking ocl-icd-libopencl1:arm64 (2.2.11-1ubuntu1) ...
Selecting previously unselected package libhwloc5:arm64.
Preparing to unpack .../19-libhwloc5_1.11.9-1_arm64.deb ...
Unpacking libhwloc5:arm64 (1.11.9-1) ...
Selecting previously unselected package libhwloc-plugins.
Preparing to unpack .../20-libhwloc-plugins_1.11.9-1_arm64.deb ...
Unpacking libhwloc-plugins (1.11.9-1) ...
Selecting previously unselected package libipmimonitoring5a.
Preparing to unpack .../21-libipmimonitoring5a_1.4.11-1.1ubuntu4.1_arm64.deb ...
Unpacking libipmimonitoring5a (1.4.11-1.1ubuntu4.1) ...
Selecting previously unselected package slurm-wlm-basic-plugins.
Preparing to unpack .../22-slurm-wlm-basic-plugins_17.11.2-1build1_arm64.deb ...
Unpacking slurm-wlm-basic-plugins (17.11.2-1build1) ...
Selecting previously unselected package slurm-client.
```

```

Preparing to unpack .../23-slurm-client_17.11.2-1build1_arm64.deb ...
Unpacking slurm-client (17.11.2-1build1) ...
Selecting previously unselected package slurmd.
Preparing to unpack .../24-slurmd_17.11.2-1build1_arm64.deb ...
Unpacking slurmd (17.11.2-1build1) ...
Selecting previously unselected package slurmctld.
Preparing to unpack .../25-slurmctld_17.11.2-1build1_arm64.deb ...
Unpacking slurmctld (17.11.2-1build1) ...
Selecting previously unselected package slurm-wlm.
Preparing to unpack .../26-slurm-wlm_17.11.2-1build1_arm64.deb ...
Unpacking slurm-wlm (17.11.2-1build1) ...
Setting up libaec0:arm64 (0.3.2-2) ...
Setting up libdbil:arm64 (0.9.0-5) ...
Setting up libhwloc5:arm64 (1.11.9-1) ...
Setting up librrd8:arm64 (1.7.0-1build1) ...
Setting up gcc-7-base:arm64 (7.5.0-3ubuntu1~18.04) ...
Setting up liblua5.1-0:arm64 (5.1.5-8.1build2) ...
Setting up freeipmi-common (1.4.11-1.1ubuntu4.1) ...
Setting up libfreeipmil6 (1.4.11-1.1ubuntu4.1) ...
Setting up ocl-icd-libopencl1:arm64 (2.2.11-1ubuntu1) ...
Setting up libslurmdb32 (17.11.2-1build1) ...
Setting up libsz2:arm64 (0.3.2-2) ...
Setting up libasan4:arm64 (7.5.0-3ubuntu1~18.04) ...
Setting up libgfortran4:arm64 (7.5.0-3ubuntu1~18.04) ...
Setting up libubsan0:arm64 (7.5.0-3ubuntu1~18.04) ...
Setting up libgcc-7-dev:arm64 (7.5.0-3ubuntu1~18.04) ...
Setting up cpp-7 (7.5.0-3ubuntu1~18.04) ...
Setting up libstdc++-7-dev:arm64 (7.5.0-3ubuntu1~18.04) ...
Setting up libipmimonitoring5a (1.4.11-1.1ubuntu4.1) ...
Setting up libhwloc-plugins (1.11.9-1) ...
Setting up libhdf5-100:arm64 (1.10.0-patch1+docs-4) ...
Setting up gcc-7 (7.5.0-3ubuntu1~18.04) ...
Setting up g++-7 (7.5.0-3ubuntu1~18.04) ...
Setting up slurm-wlm-basic-plugins (17.11.2-1build1) ...
Setting up slurmd (17.11.2-1build1) ...
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 (17.11.2-1build1) ...
Setting up slurmctld (17.11.2-1build1) ...
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 (17.11.2-1build1) ...
Processing triggers for systemd (237-3ubuntu10.25) ...

```

```

Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Synchronizing state of slurmd.service with SysV service script with
/lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable slurmd
Job for slurmd.service failed because the control process exited with error code.
See "systemctl status slurmd.service" and "journalctl -xe" for details.

```

En aquesta part s'instal·la el software MPICH:

```

Reading package lists...
Building dependency tree...
Reading state information...
The following additional packages will be installed:
  gfortran gfortran-7 hwloc-nox libgfortran-7-dev libmpich-dev libmpich12
Suggested packages:
  gfortran-doc gfortran-7-doc libgfortran4-dbg libcoarrays-dev blcr-util
  mpich-doc
The following NEW packages will be installed:
  gfortran gfortran-7 hwloc-nox libgfortran-7-dev libmpich-dev libmpich12
  mpich
0 upgraded, 7 newly installed, 0 to remove and 340 not upgraded.
Need to get 10.0 MB of archives.
After this operation, 34.7 MB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 libgfortran-7-dev
arm64 7.5.0-3ubuntu1~18.04 [300 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 gfortran-7 arm64
7.5.0-3ubuntu1~18.04 [7465 kB]
Get:3 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 gfortran arm64
4:7.4.0-1ubuntu2.3 [1344 B]
Get:4 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 hwloc-nox arm64
1.11.9-1 [143 kB]
Get:5 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 libmpich12 arm64
3.3~a2-4 [740 kB]
Get:6 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 libmpich-dev arm64
3.3~a2-4 [1196 kB]
Get:7 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 mpich arm64 3.3~a2-4
[160 kB]
debconf: unable to initialize frontend: Dialog
debconf: (Dialog frontend will not work on a dumb terminal, an emacs shell buffer, or
without a controlling terminal.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (This frontend requires a controlling tty.)
debconf: falling back to frontend: Teletype
dpkg-preconfigure: unable to re-open stdin:
Fetched 10.0 MB in 1s (7300 kB/s)
Selecting previously unselected package libgfortran-7-dev:arm64.
(Reading database ... 149174 files and directories currently installed.)
Preparing to unpack .../0-libgfortran-7-dev_7.5.0-3ubuntu1~18.04_arm64.deb ...

```



```

Unpacking libgfortran-7-dev:arm64 (7.5.0-3ubuntu1~18.04) ...
Selecting previously unselected package gfortran-7.
Preparing to unpack .../1-gfortran-7_7.5.0-3ubuntu1~18.04_arm64.deb ...
Unpacking gfortran-7 (7.5.0-3ubuntu1~18.04) ...
Selecting previously unselected package gfortran.
Preparing to unpack .../2-gfortran_4%3a7.4.0-1ubuntu2.3_arm64.deb ...
Unpacking gfortran (4:7.4.0-1ubuntu2.3) ...
Selecting previously unselected package hwloc-nox.
Preparing to unpack .../3-hwloc-nox_1.11.9-1_arm64.deb ...
Unpacking hwloc-nox (1.11.9-1) ...
Selecting previously unselected package libmpich12:arm64.
Preparing to unpack .../4-libmpich12_3.3~a2-4_arm64.deb ...
Unpacking libmpich12:arm64 (3.3~a2-4) ...
Selecting previously unselected package libmpich-dev.
Preparing to unpack .../5-libmpich-dev_3.3~a2-4_arm64.deb ...
Unpacking libmpich-dev (3.3~a2-4) ...
Selecting previously unselected package mpich.
Preparing to unpack .../6-mpich_3.3~a2-4_arm64.deb ...
Unpacking mpich (3.3~a2-4) ...
Setting up hwloc-nox (1.11.9-1) ...
Setting up libmpich12:arm64 (3.3~a2-4) ...
Setting up libgfortran-7-dev:arm64 (7.5.0-3ubuntu1~18.04) ...
Setting up gfortran-7 (7.5.0-3ubuntu1~18.04) ...
Setting up gfortran (4:7.4.0-1ubuntu2.3) ...
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 mpich (3.3~a2-4) ...
update-alternatives: using /usr/bin/mpirun.mpich to provide /usr/bin/mpirun (mpirun)
in auto mode
Setting up libmpich-dev (3.3~a2-4) ...
update-alternatives: using /usr/include/mpich to provide /usr/include/mpi (mpi) in auto
mode

```

A partir d'aquí fins al final es pot veure com s'instal·la el nou idioma i es configura a més del *layout* del teclat:

```

Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Generating locales (this might take a while)...
    es_ES.UTF-8... done
Generation complete.
Reading package lists...
Building dependency tree...
Reading state information...
The following additional packages will be installed:
    language-pack-es-base language-pack-gnome-es-base
The following NEW packages will be installed:

```

```

language-pack-es language-pack-es-base language-pack-gnome-es
language-pack-gnome-es-base wspanish
0 upgraded, 5 newly installed, 0 to remove and 340 not upgraded.
Need to get 8676 kB of archives.
After this operation, 35.9 MB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 language-pack-es-
base all 1:18.04+20180712 [2845 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 language-pack-es
all 1:18.04+20200702 [2002 kB]
Get:3 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 language-pack-
gnome-es-base all 1:18.04+20180712 [2879 kB]
Get:4 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 language-pack-
gnome-es all 1:18.04+20200702 [739 kB]
Get:5 http://ports.ubuntu.com/ubuntu-ports bionic/main arm64 wspanish all 1.0.27 [211
kB]
debconf: unable to initialize frontend: Dialog
debconf: (Dialog frontend will not work on a dumb terminal, an emacs shell buffer, or
without a controlling terminal.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (This frontend requires a controlling tty.)
debconf: falling back to frontend: Teletype
dpkg-preconfigure: unable to re-open stdin:
Fetched 8676 kB in 1s (8074 kB/s)
Selecting previously unselected package language-pack-es-base.
(Reading database ... 149328 files and directories currently installed.)
Preparing to unpack .../language-pack-es-base_1%3a18.04+20180712_all.deb ...
Unpacking language-pack-es-base (1:18.04+20180712) ...
Selecting previously unselected package language-pack-es.
Preparing to unpack .../language-pack-es_1%3a18.04+20200702_all.deb ...
Unpacking language-pack-es (1:18.04+20200702) ...
Replacing files in old package language-pack-es-base (1:18.04+20180712) ...
Selecting previously unselected package language-pack-gnome-es-base.
Preparing to unpack .../language-pack-gnome-es-base_1%3a18.04+20180712_all.deb ...
Unpacking language-pack-gnome-es-base (1:18.04+20180712) ...
Selecting previously unselected package language-pack-gnome-es.
Preparing to unpack .../language-pack-gnome-es_1%3a18.04+20200702_all.deb ...
Unpacking language-pack-gnome-es (1:18.04+20200702) ...
Replacing files in old package language-pack-gnome-es-base (1:18.04+20180712) ...
Selecting previously unselected package wspanish.
Preparing to unpack .../wspanish_1.0.27_all.deb ...
Unpacking wspanish (1.0.27) ...
Setting up wspanish (1.0.27) ...
debconf: unable to initialize frontend: Dialog
debconf: (Dialog frontend will not work on a dumb terminal, an emacs shell buffer, or
without a controlling terminal.)
debconf: falling back to frontend: Readline
Setting up language-pack-es (1:18.04+20200702) ...

```

```

Setting up language-pack-es-base (1:18.04+20180712) ...
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.
Setting up language-pack-gnome-es (1:18.04+20200702) ...
Setting up language-pack-gnome-es-base (1:18.04+20180712) ...
Processing triggers for cracklib-runtime (2.9.2-5build1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for dictionaries-common (1.27.2) ...
debconf: unable to initialize frontend: Dialog
debconf: (Dialog frontend will not work on a dumb terminal, an emacs shell buffer, or
without a controlling terminal.)
debconf: falling back to frontend: Readline
Processing triggers for bamfdaemon (0.5.3+18.04.20180207.2-0ubuntu1) ...
Rebuilding /usr/share/applications/bamf-2.index...

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Reading package lists...
Building dependency tree...
Reading state information...
The following additional packages will be installed:
  xorgxrdp
Suggested packages:
  guacamole xrdp-pulseaudio-installer
The following NEW packages will be installed:
  xorgxrdp xrdp

```

```

0 upgraded, 2 newly installed, 0 to remove and 340 not upgraded.
Need to get 443 kB of archives.
After this operation, 3132 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 xorgxrdp arm64 0.9.5-2 [68.7 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 xrdp arm64 0.9.5-2 [374 kB]
debconf: unable to initialize frontend: Dialog
debconf: (Dialog frontend will not work on a dumb terminal, an emacs shell buffer, or without a controlling terminal.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (This frontend requires a controlling tty.)
debconf: falling back to frontend: Teletype
dpkg-preconfigure: unable to re-open stdin:
Fetched 443 kB in 0s (1141 kB/s)
Selecting previously unselected package xorgxrdp.
(Reading database ... 150663 files and directories currently installed.)
Preparing to unpack .../xorgxrdp_0.9.5-2_arm64.deb ...
Unpacking xorgxrdp (0.9.5-2) ...
Selecting previously unselected package xrdp.
Preparing to unpack .../xrdp_0.9.5-2_arm64.deb ...
Unpacking xrdp (0.9.5-2) ...
Setting up xrdp (0.9.5-2) ...

Generating 2048 bit rsa key...

ssl_gen_key_xrdp1 ok

saving to /etc/xrdp/rsakeys.ini

debconf: unable to initialize frontend: Dialog
debconf: (Dialog frontend will not work on a dumb terminal, an emacs shell buffer, or without a controlling terminal.)
debconf: falling back to frontend: Readline
Created symlink /etc/systemd/system/multi-user.target.wants/xrdp-sesman.service → /lib/systemd/system/xrdp-sesman.service.
Created symlink /etc/systemd/system/multi-user.target.wants/xrdp.service → /lib/systemd/system/xrdp.service.
Setting up xorgxrdp (0.9.5-2) ...
Processing triggers for systemd (237-3ubuntu10.25) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
rm: cannot remove '/home/odroid/Documents/init_slave.sh': No such file or directory

```

5.3 Execució de prova sobre el clúster de codi paral·lelitzat amb MPI

Per poder comprovar clarament que aquest software funciona correctament, s'ha fet una execució sobre el clúster, després d'haver-ho instal·lat. Que aquesta execució funcioni, valida quasi totes les instal·lacions i configuracions, ja que aquest projecte es comporta com unes fitxes de dominó, on quan cau una cauen totes.

Primer s'ha de compilar el software de prova:

```
odroid@odroid:~/Documents$ mpicc P2.2_CPM_DavidFerrer.c -O3
```

Figura 13: Compilació de codi de prova al clúster.

Execució amb 2 subtasques:

```
odroid@odroid:~/Documents$ time srun -n2 a.out
numero solucions : 13889280

real    1m23.330s
user    0m0.020s
sys     0m0.010s
```

Figura 14: Execució de codi de prova amb 2 subtasques.

Execució amb 4 subtasques:

```
odroid@odroid:~/Documents$ time srun -n4 -N2 a.out
numero solucions : 13889280

real    0m42.279s
user    0m0.040s
sys     0m0.000s
```

Figura 15: Execució de codi de prova amb 4 subtasques.

Execució amb 8 subtasques:

```
odroid@odroid:~/Documents$ time srun -n8 -N2 a.out
numero solucions : 13889280

real    0m21.322s
user    0m0.020s
sys     0m0.010s
```

Figura 16: Execució de codi de prova amb 8 subtasques.

Podem observar com en tots els casos, el software de prova s'executa correctament i per cada increment en les subtasques, es redueix el temps d'execució. Complint d'aquesta manera l'objectiu final d'un servidor paral·lel i distribuït.

6 Conclusions

Després d'haver finalitzat el treball, els meus coneixements sobre sistemes distribuïts i paral·lels apresos a les assignatures de Arquitectura de computadors i Computació paral·lela i massiva han estat posats en practica, sobre un sistema real, i d'aquesta manera he aprofundit en els coneixements d'aquests, que d'altra manera no hagués après, i per tant no hagués aconseguit un dels meus objectius que era aprendre més sobre el tema.

D'altra banda també he pogut aprendre més sobre l'assignatura de Gestió de Sistemes i Xarxes ja que he tingut l'oportunitat de poder treballar en un sistema no practicat en aquesta assignatura i aprendre noves tecnologies per a mi, com *dnsmasq*, i a l'hora he pogut obtindre una visió més amplia sobre l'automatització de sistemes, veient els inconvenient i les avantatges de treballar sobre aquests, donant-me l'oportunitat de veure com seria treballar en aquest camp.

Per un altre costat m'ha servit per tindre en compte la dificultat de treballar en programes més grans dels que es poden veure durant la carrera, i per aquest motiu valorar més el treball en equip.

Durant el desenvolupament del projecte m'he adonat de la dificultat de treballar en un software que executa accions de manera remota, ja que s'han de tindre en compte molts aspectes relacionats amb la seguretat, que d'altra manera no m'havia arribat a plantejar, al no trobar-me davant d'aquesta situació.

Per finalitzar aquest apartat, he pogut experimentar per mi mateix la dificultat de treballar sobre Linux, ja que s'han de tindre en compte moltíssims factors que poden esdevenir al fer petits canvis, ja que tot està relacionat entre si.

7 Recursos utilitzats

- [1] Pàgina Web https://www.sdcard.org/about_sda/index.html .[consulta] 1/07/2020
- [2] Pàgina Web <https://www.hardkernel.com/shop/odroid-c2/> .[consulta] 3/07/2020
- [3] Pàgina Web https://wiki.odroid.com/accessory/emmc/reference_chart .[consulta] 3/07/2020
- [4] Pàgina Web <https://developer.arm.com/ip-products/graphics-and-multimedia/mali-gpus/mali-450-gpu> .[consulta] 3/08/2020
- [5] Pàgina Web <https://manpages.debian.org/testing/slurm-client/slurm-wlm.1> .[consulta] 20/08/2020
- [6] Pàgina Web https://www.youtube.com/watch?v=NH_Fb7X6Db0&feature=relmfu .[consulta] 25/08/2020
- [7] Pàgina Web <https://slurm.schedmd.com/slurm.conf.html> .[consulta] 5/09/2020
- [8] Pàgina Web <https://slurm.schedmd.com/quickstart.html> .[consulta] 20/09/2020
- [9] Pàgina Web [https://en.wikipedia.org/wiki/Master/slave_\(technology\)](https://en.wikipedia.org/wiki/Master/slave_(technology)) .[consulta] 12/08/2020
- [10] Pàgina Web https://slurm.schedmd.com/SUG14/sched_tutorial.pdf .[consulta] 25/09/2020
- [11] Pàgina Web https://commons.wikimedia.org/wiki/File:Auswirkungen_von_FIFO_und_Backfill_Strategie_n.png .[consulta] 27/10/2020
- [12] Pàgina Web <https://linux.die.net/man/7/munge> .[consulta] 30/08/2020
- [13] Pàgina Web <https://slurm.schedmd.com/download.html> .[consulta] 26/09/2020
- [14] Pàgina Web <https://www.electronics-notes.com/articles/connectivity/ethernet-ieee-802-3/cables-types-pinout-cat-5-5e-6.php> .[consulta] 16/07/2020
- [15] Pàgina Web <https://es.slideshare.net/lilianaalfonso/topologas-de-red-switchhubrouter> .[consulta] 18/07/2020
- [16] Pàgina Web <https://askubuntu.com/questions/328240/assign-vnc-password-using-script> .[consulta] 01/11/2020
- [17] Pàgina Web <https://www.imd.guru/sistemas/bash/ifs.html> .[consulta] 09/09/2020
- [18] Pàgina Web <https://www.computerhope.com/unix/bash/read.htm> .[consulta] 09/09/2020
- [19] Pàgina Web <http://manpages.ubuntu.com/manpages/bionic/es/man1/debconf-set-selections.1.html> .[consulta] 04/10/2020
- [20] Pàgina Web <http://ecar2017.hpclatam.org/files/slurm.pdf> .[consulta] 23/09/2020
- [21] Pàgina Web <https://computingforgeeks.com/how-to-install-vnc-server-on-ubuntu-18-04-lts/> .[consulta] 01/08/2020

8 Annexes

8.1 Manual d'instal·lació

Aquest apartat dels annexes, té la intenció de simular un manual per a la correcta instal·lació del projecte per així, d'una manera automatitzada poder dur a terme el muntatge del clúster. La instal·lació tindrà dos camins, és a dir, l'usuari podrà escollir si vol una interfície gràfica o no. Sense més preàmbuls, a continuació es podrà veure el manual.

1. Primer de tot s'haurà d'accedir a la placa *master* mitjançant la comanda *ssh nom@ip*.
2. Per tant, en un ordinador situat a la mateixa xarxa, se li obrirà una terminal i executarem la comanda anterior on nom és *Odroid* i ip és l'adreça IP de la placa *master*.
3. A continuació, se'ns demanarà una contrasenya. Si és el primer cop en entrar a la placa, la contrasenya per defecte serà *odroid*. En cas contrari, us haureu d'enrecordar de la contrasenya que vau posar si la vàreu canviar.
4. Un cop dintre, ens situarem dintre del directori *Documents/* amb l'ajuda de la comanda *cd Documents/* i seguidament s'executarà un update amb la comanda *sudo apt update -y*. Se'ns tornarà a demanar la contrasenya que per defecte serà *odroid*.
5. Finalment, se'ns mostrarà un error degut a que les imatges que proporciona Hardkernel, els hi falta una clau per accedir a un dels seus repositoris. Aquest error es solucionarà amb l'execució del codi clonat. Per a clonar el codi s'ha d'executar la comanda *git clone <https://github.com/davidf2/odroid-cluster.git>*.
6. Per acabar anirem al directori clonat executant la comanda *cd odroid-cluster*.
7. Un cop dintre tindrem dos opcions, la opció a) que serà l'execució del codi de manera remota per SSH. O l'opció b) que serà la desconnexió del SSH i entrar per VNC o per un monitor connectat per HDMI per tal de veure la interfície gràfica.
 - a. Simplement, s'executarà la comanda *sudo ./init_master.sh* i començarà el procés d'instal·lació automàtic. Un cop finalitzat el procés ja tindrem el nostre sistema de places *Odroid* muntat i amb un correcte funcionament.
 - b. Per executar aquest codi mitjançant la interfície gràfica s'ha d'utilitzar la comanda *sudo python3 installer_gui.pyw*, i pitjar el botó *INSTALL NOW*, això obrirà una finestra per a que l'usuari introdueixi la contrasenya, i tot seguit es mostrarà una terminal amb l'execució del programa.

8.2 Codi

8.2.1 *init_master.sh*

```
#!/bin/bash
if [ "$EUID" -ne 0 ]
  then echo "Please run as root"
  exit
fi
# Deshabilitem Unattended-Upgrade
systemctl disable unattended-upgrades
systemctl stop unattended-upgrades
apt remove unattended-upgrades -y
cp -p odroid_cluster.conf /etc
```



```

cp -p network_lib.sh /usr/local/sbin/
# Carreguem els scripts network_lib.sh i locale.sh com a llibreries, per
# poder fer servir les seves funcions
source network_lib.sh
source ./locale.sh
scripts_path="$(cat /etc/odroid_cluster.conf | grep "^SCRIPTS_DIR=" | cut -d= -f2)"
externaldns1="$(cat /etc/odroid_cluster.conf | grep "^EXTERNALDNS1=" | cut -d= -f2)"
externaldns2="$(cat /etc/odroid_cluster.conf | grep "^EXTERNALDNS2=" | cut -d= -f2)"
upgrade="$(cat /etc/odroid_cluster.conf | grep "^UPGRADE=" | cut -d= -f2)"
# Agafem el nom de l'usuari no root
master_name=$(cat /etc/odroid_cluster.conf | grep "^DEFAULT_USER=" | cut -d= -f2)
# Agafem el directori home l'usuari no root
master_home=$(eval echo "~$master_name")
PUB_KEY_FILE="${master_home}/.ssh/id_rsa.pub"
KEY_FILE=$(echo $PUB_KEY_FILE | cut -d. -f1-2)
KNOWN_HOSTS="${master_home}/.ssh/known_hosts"
locale="$(cat /etc/odroid_cluster.conf | grep "^SYS_LANGUAGE=" | cut -d= -f2)"
layout="$(cat /etc/odroid_cluster.conf | grep "^LAYOUT=" | cut -d= -f2)"
variant="$(cat /etc/odroid_cluster.conf | grep "^VARIANT=" | cut -d= -f2)"
change_password() {
    pass="0"
    pass2="1"
    # Demana la contrasenya dos cops, fins que coincideixin
    while [ $pass != $pass2 ]; do
        echo "Enter the new password for the master node:"
        stty -echo
        read -r pass
        stty sane
        echo "Re-enter the new password:"
        stty -echo
        read -r pass2
        stty sane
    done
    # Modifica la contrasenya de l'usuari root i el per defecte
    echo -e "${pass}\n${pass}" | passwd &> /dev/null
    echo -e "${pass}\n${pass}" | passwd $master_name &> /dev/null
    # Modifiquem el password de vnc
    #./set_vnc_password.sh "$pass"
}
add_ssh() {
    # Instal·lem openssh server
    apt-get install openssh-server sshpass -y
    # Iniciem el servei ssh
    systemctl start sshd
    systemctl enable sshd
    # Si no existeixen generem el parell de claus
    if [ ! -f "$KEY_FILE" ]; then
        su $master_name -c "ssh-keygen -q -t rsa -f \"$KEY_FILE\" -N
        \"\$passphrase\""
        fi
        #Ens autoafegim a knownhosts, per a quan estigui el servidor NFS /home
        su $master_name -c "echo \"$(ssh-keyscan -H $(hostname))\" >> $KNOWN_HOSTS"
        #Ens autoafegim la clau publica, per a quan estigui el servidor NFS /home
        cat "$PUB_KEY_FILE" > "$master_home"/.ssh/authorized_keys
        # Si no existeix, fem una còpia del fitxer de configuració
        # original del servidor ssh
        if [[ ! -f /etc/ssh/sshd_config.back ]]; then
            bash -c "cp /etc/ssh/sshd_config /etc/ssh/sshd_config.back"
        fi
        # Carreguem la configuració per al dimoni sshd
        echo "#Port 5000"
        # No permetre fer login com a root
        PermitRootLogin no
        # Fer servir la versió 2 de ssh, més segura que la 1
        Protocol 2
        PermitEmptyPasswords no
        ChallengeResponseAuthentication no
        UsePAM no
        GatewayPorts yes
        X11Forwarding yes
        PrintMotd no
        # Allow client to pass locale environment variables
        AcceptEnv LANG LC_*
        # override default of no subsystems
        Subsystem sftp /usr/lib/openssh/sftp-server" > /etc/ssh/sshd_config
        # Reiniciem el dimoni de ssh per a que carregui la nova configuració

```

```

systemctl restart sshd
#apt-get install fail2ban -y
#cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
#systemctl restart fail2ban
}
add_dnsmasq() {
    if [ $# -lt 1 ]; then
        echo -e "Error, at least you have to enter 1 parameter, for more information
\n\t init_master -h"
        exit 1
    fi
    ip="$1"
    lan_interface="$2"
    mask="$3"
    # Instal·lem dnsmasq i el deshabilitem
    apt-get install dnsmasq -y 2> /dev/null
    systemctl disable --now dnsmasq
    # Deshabilitem el dimoni systemd-resolved per a que no canviï la configuració del
DNS
    systemctl disable systemd-resolved
    systemctl stop systemd-resolved
    ifdown "$lan_interface"
    ifup "$lan_interface"

    # Descomentem
    sed -i '/prepend domain-name-servers 127.0.0.1;/s/^#//g' /etc/dhcp/dhclient.conf
    # Si no existeix fem una còpia de seguretat del fitxer dnsmasq.conf
    if [ [ ( ! -f /etc/dnsmasq.conf.back ) && ( -f /etc/dnsmasq.conf ) ] ]; then
        bash -c "cp -p /etc/dnsmasq.conf > /etc/dnsmasq.conf.back"
    fi
    first_ip=$(calculate_first_ip "$ip" "$mask")
    last_ip=$(calculate_last_ip "$ip" "$mask")
    # Carreguem la configuració per a dnsmasq
    echo "
listen-address=::1,127.0.0.1,${ip}
domain-needed
bogus-priv
no-hosts
hostsdir=/etc/hosts.d
strict-order
no-resolv
cache-size=1000

server=${externaldns1}
server=${externaldns2}

domain=lan
#local=/lan/

interface=${lan_interface}
dhcp-range=${first_ip},${last_ip},12h
# Establecer la porta de enllac predeterminada.
dhcp-option=option:router,${ip}
# Establecer servidors DNS para anunciar
dhcp-option=option:dns-server,${ip}
dhcp-script=${scripts_path}/dhcp_script.sh
" > /etc/dnsmasq.conf
    # Reiniciem i habilitem el dimoni de dnsmasq
    systemctl enable dnsmasq
    systemctl start dnsmasq
}
}
add_vnc() {
    # Instal·lem l'entorn d'escriptori Xfce
    apt-get install xfce4 xfce4-goodies -y
    # Instal·lem un servidor VNC
    apt-get install tightvncserver -y
    # Carreguem la configuració de VNC pel nou entorn d'escriptori
    echo "
#!/bin/bash
exec /usr/bin/startxfce4 &
" > "$master_home"/.vnc/xstartup
}
}
add_nfs() {
    if [ $# -lt 2 ]; then
        echo "Error, you must enter 2 parameters, the first one corresponding to

```

```

        an IP and the second one to the mask"
        exit 1
    fi
    ip="$1"
    mask="$2"
    apt-get install nfs-kernel-server -y
    ip_net="$(calculate_network_ip $ip $mask)"
    mask_cidr="$(mask_to_cidr $mask)"
    echo "/home ${ip_net}${mask_cidr}(rw,no_root_squash,no_subtree_check)" >>
/etc/exports
    exportfs -arv
    systemctl enable nfs-kernel-server
    systemctl restart nfs-kernel-server
}
add_munge() {
    apt-get install munge -y
    systemctl enable --now munge
    /usr/sbin/create-munge-key -f
    systemctl restart munge
    # Copiem la clau de munge a /home, per a que puguin copiarla els slaves
    dd if=/etc/munge/munge.key of=/home/munge.key
}
clean_tmp_hosts() {
    # Afegim un nou servei que s'encarrega de netejar el fitxer de
    # hosts temporal al tancar el sistema
    echo "[Unit]
Description=Clean /etc/hosts.d/tmp_hosts file
DefaultDependencies=no
Before=shutdown.target
[Service]
Type=oneshot
ExecStart=/bin/sh -c 'echo "" > /etc/hosts.d/tmp_hosts'
TimeoutStartSec=0
[Install]
WantedBy=shutdown.target" > /etc/systemd/system/clean_tmp_hosts.service

    systemctl enable clean_tmp_hosts.service
}
add_monitoring() {
    apt-get install git -y
    cd $(eval echo ~$name)/Downloads
    git clone https://github.com/JoanJaraBosch/TFG.git
    chown -R odroid: TFG/
    cd TFG
    ./start-monitoring.sh
}
# Creem el directori principal, on emmagatzemarem els scripts necessaris
if [ ! -d "$scripts_path" ]; then
    mkdir "$scripts_path"
fi
# Copiem el fitxer que s'executa cada cop que el servidor dhcp fa una modificació
cp -p dhcp_script.sh "$scripts_path"/
# Copiem els scripts dependents
cp -p init_slave.sh "$scripts_path"/
cp -p add_slave.sh "$scripts_path"/
cp -p locale.sh "$scripts_path"/
cp -p iptables.sh "$scripts_path"/
# Creem el directori on guardarem alguns logs
mkdir /var/log/odroid_cluster
chown "$master_name": /var/log/odroid_cluster
# Solucionem error de claus amb l'update
apt-key adv -v --keyserver keyserver.ubuntu.com --recv-keys 5360FB9DAB19BAC9
# Actualitzem el master
apt-get update -y
# Modifiquem el hostname a master
hostnamectl set-hostname master
# Modifiquem el teclat
set_layout "$layout" "$variant"
apt install expect -y
# Obliguem a l'usuari a canviar la contrasenya del master
change_password
# Fiquem a zona horaria i actualitzem l'hora
timedatectl set-timezone "$(cat /etc/odroid_cluster.conf | grep "^SYS_TIMEZONE=" | cut -d=
-f2)"
apt-get install chrony -y

```

```

systemctl enable --now chronyd
# Evitem que el diàleg amb la GUI durant la instal·lació de iptables-persistent
echo iptables-persistent iptables-persistent/autosave_v4 boolean true | debconf-set-
selections
echo iptables-persistent iptables-persistent/autosave_v6 boolean true | debconf-set-
selections
# Iptables persistent
apt-get install iptables-persistent -y
systemctl enable netfilter-persistent
# Creem el directori de hosts compartits per a dnsmasq
mkdir /etc/hosts.d
# Cridem al script de configuració de xarxa
result=$(./conf_network_master.sh)
# Del resultat del script agafem només l'última línia i descartem les demés
echo "$result" | head -n $(expr $(echo "$result" | wc -l) - 1) 2> /dev/null
result=$(echo $result | awk -F' ' '{print $NF}')
# Guardem el resultat en un array per treballar més comodament
IFS=';' read -a net_array <<< "$result"
# Instal·lem el servidor openssh a més d'altre software relacionat, i el configurem
add_ssh
# Esborrem software innecessari
apt-get remove --purge libreoffice* thunderbird pacman transmission* mate-* -y
apt-get autoremove -y
apt-get autoclean -y
# Instal·lem VNC i l'entorn gràfic xfce4
add_vnc
add_nfs "${net_array[0]}" "${net_array[1]}"
add_munge
clean_tmp_hosts
./install_slurm.sh "${net_array[0]}" "${net_array[1]}"
apt-get install mpich -y
# Assegurem que la xarxa interna estigui en up
ifup --force "${net_array[3]}"
# Instal·lem el servidor dns i dhcp dnsmasq i el configurem
# AIXÓ SEMPRE HA DE SER L'ULTIM QUE FEM ABANS DEL UPGRADE
add_dnsmasq "${net_array[0]}" "${net_array[3]}" "${net_array[1]}"
# Instal·lem el software de monitoreig fet per Joan Jara Bosch
add_monitoring
# Modifiquem l'idioma
set_language "$locale"
if [ "$upgrade" -eq 1 ]; then
    apt-get upgrade -y
fi

```

8.2.2 *conf_network_master.sh*

```

#!/bin/bash
# Carreguem el script network_lib.sh com a una llibreria, per
# poder fer servir les seves funcions
source network_lib.sh
add_iptables() {
    net_interface="$1"
    lan_interface="$2"
    path="$3"
    # Configurem les iptables
    "$path"/iptables.sh "$lan_interface" "$net_interface"

    # Guardem els canvis a iptables de forma permanentment
    iptables-save > /etc/iptables/rules.v4
    iptables-save > /etc/iptables/rules.v6
}
# Default values
ip=$(cat /etc/odroid_cluster.conf | grep "^IP=" | cut -d= -f2)
mask=$(cat /etc/odroid_cluster.conf | grep "^MASK=" | cut -d= -f2)
class=$(cat /etc/odroid_cluster.conf | grep "^IP_CLASS=" | cut -d= -f2)
scripts_path=$(cat /etc/odroid_cluster.conf | grep "^SCRIPTS_DIR=" | cut -d= -f2)
line=$(cat /etc/hosts | grep 127.0.0.1)
host=$(hostname)
host2=$(echo $(who am i | awk '{print $1}'))
sed -i 's/^\ "$line".*/"$line" "$host" "$host2"/g' /etc/hosts
result=$(check_interfaces)
if [[ $? -ne 0 ]]; then
    exit 1
fi

```

```

net_interface=$(echo $result | cut -d ";" -f 1)
lan_interface=$(echo $result | cut -d ";" -f 2)
echo "auto lo
iface lo inet loopback
auto ${lan_interface}
iface ${lan_interface} inet static
    address ${ip}
    netmask ${mask//:/}
auto ${net_interface}
iface ${net_interface} inet dhcp" > /etc/network/interfaces
echo "$ip master" >> /etc/hosts.d/lan_hosts
# Afegim la interfície de xarxa lan al fitxer /run/network/ifstate
if [ ! $(cat /run/network/ifstate | grep "$lan_interface") ]; then
    echo "$lan_interface=$lan_interface" >> /run/network/ifstate
fi
# Afegim la interfície de xarxa internet al fitxer /run/network/ifstate
if [ ! $(cat /run/network/ifstate | grep "$net_interface") ]; then
    echo "$net_interface=$net_interface" >> /run/network/ifstate
fi
# Reiniciem la interfície de xarxa (xarxa interna)
ifdown --force $lan_interface
ifup --force $lan_interface
# Habilitem de forma permanent el forwarding, descomentant la línia pertinent
sed -i '/net.ipv4.ip_forward=1/s/^#//g' /etc/sysctl.conf
# Carrega els canvis sense reiniciar
sysctl -p
add_iptables "$net_interface" "$lan_interface" "$scripts_path"
# Retornem els resultats
echo "$ip;$mask;$net_interface;$lan_interface"

```

8.2.3 iptables.sh

```

#!/bin/bash
if [ $# -ne 2 ]; then
    echo "Error, you have to enter 2 parameters."
    exit 1
fi
cluster_lan="$1"
internet="$2"
# Esborrem les regles anteriors
iptables -F
iptables -X
iptables -t nat -F
iptables -t nat -X
iptables -t mangle -F
iptables -t mangle -X
# Afegim les polítiques per defecte
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
# Permetem les entrades i sortides de la interfície loopback
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
# Acceptem les respostes en connexions establertes
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
# Habilitem el postROUTING a iptables per donar accés a internet a la xarxa interna
iptables -t nat -A POSTROUTING -o $internet -j MASQUERADE
# Acceptem connexions SSH d'entrada i sortida només cap a la xarxa interna
iptables -A INPUT -p TCP --dport 22 -j ACCEPT
iptables -A OUTPUT -p TCP --dport 22 -o $cluster_lan -j ACCEPT
# Acceptem connexions DNS d'entrada per la xarxa interna
iptables -A INPUT -p TCP --dport 53 -i $cluster_lan -j ACCEPT
iptables -A INPUT -p UDP --dport 53 -i $cluster_lan -j ACCEPT
# Acceptem connexions DNS de sortida, pels servidors configurats
iptables -A OUTPUT -p TCP --dport 53 -j ACCEPT
iptables -A OUTPUT -p UDP --dport 53 -j ACCEPT
# Acceptem les entrades pel port 67 amb origen port 68 per al servidor DHCP
iptables -A INPUT -p UDP --dport 67 -i $cluster_lan -j ACCEPT
iptables -A OUTPUT -p UDP --dport 68 -j ACCEPT
# Acceptem les sortides HTTP i HTTPS per les actualitzacions o navegar per internet

```

```

iptables -A OUTPUT -p TCP --dport 80 -o $internet -j ACCEPT
iptables -A OUTPUT -p TCP --dport 443 -o $internet -j ACCEPT
# Fem forwarding dels paquets que surtin de la xarxa interna cap a internet, pel port 80 i
443 com a destí
iptables -A FORWARD -p TCP --dport 80 -i $cluster_lan -o $internet -j ACCEPT
iptables -A FORWARD -p TCP --dport 443 -i $cluster_lan -o $internet -j ACCEPT
# Acceptem les entrades pel port 443 pel servidor web
iptables -A INPUT -p TCP --dport 443 -i $internet -j ACCEPT
# Acceptem les entrades pel port 3000, pel servidor node.js, per la monitorització i el
manteniment
iptables -A INPUT -p TCP --dport 3000 -i $internet -j ACCEPT
# Acceptem les entrades pel port 5901 pel servidor VNC
iptables -A INPUT -p TCP --dport 5901 -i $internet -j ACCEPT
# Acceptem les sortides i entrades pel port 6817 i 6818 per a les comunicacions
# entre slurmctld i slurmd
iptables -A INPUT -p TCP --dport 6817 -i $cluster_lan -j ACCEPT
iptables -A OUTPUT -p TCP --dport 6818 -o $cluster_lan -j ACCEPT
# Acceptem la entrada TCP des de la xarxa interna, degut a la obertura de ports dinamica de
srun i mpirun
iptables -A INPUT -p TCP -i $cluster_lan -j ACCEPT
# Habilitem ICMP a la xarxa interna
iptables -A INPUT -p ICMP -i $cluster_lan -j ACCEPT
iptables -A OUTPUT -p ICMP -o $cluster_lan -j ACCEPT
# Habilitem fer ping de dins cap a fora
iptables -A OUTPUT -p ICMP --icmp-type 8 -o $internet -j ACCEPT
iptables -A FORWARD -p ICMP --icmp-type 8 -o $internet -j ACCEPT
# Acceptem les entrades pel port 2049 NFS
iptables -A INPUT -p TCP --dport 2049 -i $cluster_lan -j ACCEPT
iptables -A INPUT -p UDP --dport 2049 -i $cluster_lan -j ACCEPT
# Acceptem les sortides i el reenviament de NTP per a chrony
iptables -A OUTPUT -p UDP --dport 123 -o $internet -j ACCEPT
iptables -A FORWARD -p UDP --dport 123 -o $internet -j ACCEPT
# Acceptem les sortides i el reenviament del port 11371 per a poder fer apt-key cap a
keyserver.ubuntu.com
iptables -A OUTPUT -p TCP --dport 11371 -o $internet -j ACCEPT
iptables -A FORWARD -p TCP --dport 11371 -o $internet -j ACCEPT

```

8.2.4 *install_slurm.sh*

```

#!/bin/bash
# Carreguem el script network_api.sh com a una llibreria, per
# poder fer servir les seves funcions
source network_lib.sh
SLURM_ETC=/etc/slurm-llnl
ip=$1
mask=$2
# Instal·lem slurm-wlm
apt install slurm-wlm -y
mkdir -p "$SLURM_ETC"/cgroup
cp -p /usr/share/doc/slurmd/examples/cgroup.release_common
"$SLURM_ETC"/cgroup/cgroup.release_common
ln -s "$SLURM_ETC"/cgroup/cgroup.release_common "$SLURM_ETC"/cgroup/release_devices
ln -s "$SLURM_ETC"/cgroup/cgroup.release_common "$SLURM_ETC"/cgroup/release_cpuset
ln -s "$SLURM_ETC"/cgroup/cgroup.release_common "$SLURM_ETC"/cgroup/release_freezer
echo "/dev/null
/dev/urandom
/dev/zero
/dev/cpu/*/.*
/dev/pts/*" > "$SLURM_ETC"/allowed_devices.conf
memori=$(free --mebi | grep "Mem:" | awk '{print $2}')
cpus=$(lscpu | grep "^CPU(s):" | awk '{print $2}')
threads=$(lscpu | grep "^Thread(s)" | awk '{print $4}')
sockets=$(lscpu | grep "^Socket(s)" | awk '{print $2}')
cores=$(lscpu | grep "^Core(s)" | awk '{print $4}')
names=$(cat /etc/odroid_cluster.conf | grep "^HOSTS_NAME=" | cut -d= -f2)
max_time=$(cat /etc/odroid_cluster.conf | grep "^MAX_TIME=" | cut -d= -f2)
scheduler=$(cat /etc/odroid_cluster.conf | grep "^SCHEDULER=" | cut -d= -f2)
tracking=$(cat /etc/odroid_cluster.conf | grep "^TRACKING=" | cut -d= -f2)
echo "# slurm.conf file generated by configurator easy.html.
# Put this file on all nodes of your cluster.
# See the slurm.conf man page for more information.
#

```

```

ControlMachine=master
#
#MailProg=/bin/mail
MpiDefault=pmi2
#MpiParams=ports=#-#
ProctrackType=proctrack/cgroup
ReturnToService=1
SlurmctldPidFile=/var/run/slurm-llnl/slurmctld.pid
#SlurmctldPort=6817
SlurmdPidFile=/var/run/slurm-llnl/slurmd.pid
#SlurmdPort=6818
SlurmdSpoolDir=/var/spool/slurmd
SlurmUser=slurm
#SlurmdUser=root
StateSaveLocation=/var/spool/slurm
SwitchType=switch/none
TaskPlugin=task/cgroup
#
#
# TIMERS
#KillWait=30
#MinJobAge=300
#SlurmctldTimeout=120
#SlurmdTimeout=300
#
#
# SCHEDULING
SchedulerType=${scheduler}
SelectType=select/cons_res
SelectTypeParameters=CR_CPU
#
#
# LOGGING AND ACCOUNTING
AccountingStorageType=accounting_storage/none
ClusterName=cluster
#JobAcctGatherFrequency=30
JobAcctGatherType=jobacct_gather/none
#SlurmctldDebug=info
SlurmctldLogFile=/var/log/slurm-llnl/slurmctld.log
#SlurmdDebug=info
SlurmdLogFile=/var/log/slurm-llnl/slurmd.log
#
#
# COMPUTE NODES
NodeName=${names}[1-2] CPUs=${cpus} RealMemory=${memori} Sockets=${sockets}
CoresPerSocket=${cores} ThreadsPerCore=${threads} State=UNKNOWN
PartitionName=main Nodes=${names}[1-2] Default=YES MaxTime=${max_time} State=UP" >
"${SLURM_ETC}/slurm.conf
echo "CgroupAutomount=yes
CgroupReleaseAgentDir=\"${SLURM_ETC}/cgroup\"
ConstrainCores=yes
TaskAffinity=yes
ConstrainDevices=yes
AllowedDevicesFile=\"${SLURM_ETC}/allowed_devices.conf\"
ConstrainRAMSpace=no" > "${SLURM_ETC}/cgroup.conf
chown -R slurm: "${SLURM_ETC}"
if [ $(cat /etc/exports | grep "${SLURM_ETC}" | wc -l) -eq 0 ]; then
    echo "${SLURM_ETC} $(calculate_network_ip $ip $mask)$(mask_to_cidr
$mask) (rw,no_root_squash,no_subtree_check)" >> /etc/exports
fi
exportfs -a
# Creem un nou directori dintre de /var/spool anomenat slurm per a que slurmctld pugui fer-
lo servir
mkdir /var/spool/slurm
chown slurm: /var/spool/slurm
systemctl enable slurmctld
systemctl start slurmctld

```

8.2.5 *set_vnc_password.sh*

```
#!/bin/bash
pass="$1"
prog=/usr/bin/vncpasswd
/usr/bin/expect <<EOF
spawn "$prog"
expect "Password:"
send "$pass\r"
expect "Verify:"
send "$pass\r"
expect "Would you like to enter a view-only password (y/n)?"
send "n\r"
expect eof
exit
EOF
```

8.2.6 *network_lib.sh*

```
#!/bin/bash
# Funció per controlar que una IP estigui en un format correcte
check_ip() {
    if [[ $(echo $1 | grep [^0-9.]) ]]; then
        echo "ERROR: Incorrect IP format, can only contain numbers and dots" 1>&2
        return 1
    fi
    if [[ $(echo $1 | awk -F'.' '{print NF}') -ne 4 ]]; then
        echo "ERROR: Incorrect IP format, incorrect octets number" 1>&2
        return 1
    fi
    if [[ $(echo $1 | cut -d "." -f 4) -eq 255 ]]; then
        echo "ERROR: You cannot put broadcast address as the master's IP" 1>&2
        return 1
    fi
    octet=1
    while [ $(echo $1 | cut -d "." -f $octet) ]; do
        if [[ $(echo $1 | cut -d "." -f $octet) -lt 0 ]] || [[ $(echo $1 | cut -d
"." -f $octet) -gt 255 ]]; then
            echo "ERROR: Incorrect IP format $(echo $1 | cut -d "." -f $octet)"
            return 1
        fi
        let octet=octet+1
    done
    case $(echo $1 | cut -d "." -f 1) in
        10)
            result="A" # IP privada de classe A
            return 0
            ;;
        172)
            if [[ $(echo $1 | cut -d "." -f 2) -ge 16 ]] && [[ $(echo $1 | cut -d
"." -f 2) -le 31 ]]; then
                return 0
            fi
            ;;
        192)
            if [[ $(echo $1 | cut -d "." -f 2) -eq 168 ]]; then
                return 0
            fi
            ;;
    esac
    return 1
}
# Funció per controlar que la mascara de xarxa sigui correcta
check_mask() {
    mask=$1
    class=$(cat /etc/odroid_cluster.conf | grep "^IP_CLASS=" | cut -d= -f2)
    if [[ $(echo $mask | grep [^0-9:]) ]]; then
        echo "ERROR: Incorrect mask format, can only contain numbers and colons"
        return 1
    fi
}
```



```

fi
if [[ $(echo $mask | awk -F':' '{print NF}') -ne 4 ]]; then
    echo "ERROR: Incorrect mask format, incorrect octets number" 1>&2
    return 1
fi
case $class in
    A) min=1;;
    B) min=2;;
    C) min=2;;
esac
octet=1
while [ $(echo $mask | cut -d ":" -f $octet) ]; do
    if [[ $octet -le $min ]]; then
        if [[ $(echo $1 | cut -d ":" -f $octet) -ne 255 ]]; then
            echo "ERROR: Incorrect mask format" 1>&2
            return 1
        fi
    else
        if [[ $(echo $mask | cut -d ":" -f $octet) -lt 0 ]] || [[ $(echo
$mask | cut -d ":" -f $octet) -gt 255 ]]; then
            echo "ERROR: Incorrect mask format" 1>&2
            return 1
        fi
    fi
    let octet=octet+1
done
}
# El primer parametre retorna la interfície de xarxa amb connexió a internet
check_interfaces() {
    iptables -t nat -D POSTROUTING 1
    OLDIFS=$IFS
    IFS=$'\t\n'
    # Cerca la NIC amb connexió a internet
    for nic in $(echo $(sed 'ld;2d' /proc/net/dev | grep -v 'lo' | cut -d: -f1)); do
        if [[ $(ping 8.8.8.8 -I $nic -w2 2> /dev/null | grep "received" | cut -d " "
-f4) -gt 0 ]]; then
            net_interface="$nic"
        else
            lan_interface="$nic"
        fi
    done
    IFS=$OLDIFS
    # Comprovació final
    if [[ -z "$net_interface" ]]; then
        echo "ERROR: No internet connected interfaces found" 1>&2
        return 1
    fi
    # Comprovació final
    if [[ -z "$lan_interface" ]]; then
        echo "ERROR: Not found the second network interface" 1>&2
        return 1
    fi
    echo "$net_interface;$lan_interface"
}
# Funció que retorna una ip, segons una interfície de xarxa pasada per parametre.
get_ip_of_nic() {
    if [ $# -lt 1 ]; then
        echo "Error, you have to enter 1 parameter corresponding to the network
interface."
        exit 1
    fi
    interface="$1"

    nic_ip=""
    nic_info=$(ip -4 a show $interface)
    OLDIFS=$IFS
    IFS=$' '
    for ip in $(echo $(hostname -I)); do
        if [ $(echo "$nic_info" | grep "$ip" | wc -l) -gt 0 ]; then
            nic_ip="$ip"
        fi
    done
    IFS=$OLDIFS
    if [ "$nic_ip" == "" ]; then
        return 1
    fi
}

```

```

        fi
        echo "$nic_ip"
    }
mask_to_cidr() {
    if [ $# -lt 1 ]; then
        echo "Error, you have to enter 1 parameter corresponding to the mask."
        exit 1
    fi
    mask="$1"
    check_mask "$mask"
    result=$?
    if [[ $result -eq 0 ]]; then
        result=0
        IFS=: read -a mask_array <<< "$mask"
        for i in "${mask_array[@]}; do
            number=$(echo "obase=2;$i" | bc | awk -F "1" '{print NF-1}')
            result=$((result + $number))
        done
    else
        return 1
    fi
    echo "/$result"
}
calculate_network_ip() {
    if [ $# -lt 2 ]; then
        echo "Error, you must enter 2 parameters, the first one corresponding to
        an IP and the second one to the mask"
        exit 1
    fi
    ip="$1"
    mask="$2"

    check_ip "$ip"
    result1=$?
    check_mask "$mask"
    result2=$?
    if [ "$result1" -eq 0 ] && [ "$result2" -eq 0 ]; then
        IFS=. read -r i1 i2 i3 i4 <<< "$ip"
        IFS=: read -r m1 m2 m3 m4 <<< "$mask"
        result=$(echo "$((i1 & m1)).$((i2 & m2)).$((i3 & m3)).$((i4 & m4))")
    else
        return 1
    fi
    echo $result
}
calculate_first_ip() {
    if [ $# -lt 2 ]; then
        echo "Error, you must enter 2 parameters, the first one corresponding to
        an IP and the second one to the mask"
        exit 1
    fi
    ip="$1"
    mask="$2"
    check_ip "$ip"
    result1=$?
    check_mask "$mask"
    result2=$?
    if [ "$result1" -eq 0 ] && [ "$result2" -eq 0 ]; then
        IFS=. read -r i1 i2 i3 i4 <<< "$ip"
        IFS=: read -r m1 m2 m3 m4 <<< "$mask"
        result=$(echo "$((i1 & m1)).$((i2 & m2)).$((i3 & m3)).$(((i4 & m4)+1))")
    else
        return 1
    fi
    echo $result
}
calculate_last_ip() {
    if [ $# -lt 2 ]; then
        echo "Error, you must enter 2 parameters, the first one corresponding to
        an IP and the second one to the mask"
        exit 1
    fi
    ip="$1"

```

```

mask="$2"
check_ip "$ip"
result1=$?
check_mask "$mask"
result2=$?
if [ "$result1" -eq 0 ] && [ "$result2" -eq 0 ]; then
    IFS=. read -r i1 i2 i3 i4 <<< "$ip"
    IFS=: read -r m1 m2 m3 m4 <<< "$mask"
    result=$(echo "$((i1 & m1 | 255-m1)).$((i2 & m2 | 255-m2)).$((i3 & m3 | 255-
m3)).$((i4 & m4 | 255-m4)-1))")
    else
        return 1
    fi
echo $result
}

```

8.2.7 locale.sh

```

#!/bin/bash
set_language() {

    locale="$1"
    if [ $# -ne 1 ]; then
        echo "You need to enter a language in odroid_cluster.conf"
        exit 1
    fi
    if [ $(cat /usr/share/i18n/SUPPORTED | grep ^"$locale".UTF-8 | wc -l) -eq 0 ]; then
        echo "Incorrect language"
        exit 1
    fi
    # Instal.lem el nou idioma
    locale-gen "$locale".utf8
    # Seleccionem el nou idioma
    #update-locale LANG="$locale".UTF-8 LANGUAGE
    localectl set-locale LANG="$locale".UTF-8 LANGUAGE="$locale".UTF-8:"$(echo $locale
| cut -d_ -f1)"
    # Actualitza les variables LANG i LANGUAGE
    source /etc/default/locale
    if [ $(grep "source /etc/default/locale" /etc/profile | wc -l) -eq 0 ]; then
        echo "source /etc/default/locale" >> /etc/profile
    fi
    if [ $(grep "source /etc/default/locale" /etc/bash.bashrc | wc -l) -eq 0 ]; then
        echo "source /etc/default/locale" >> /etc/bash.bashrc
    fi
    if [ $(grep "source /etc/default/locale" /root/.profile | wc -l) -eq 0 ]; then
        echo "source /etc/default/locale" >> /root/.profile
    fi
    if [ $(grep "source /etc/default/locale" /root/.bashrc | wc -l) -eq 0 ]; then
        echo "source /etc/default/locale" >> /root/.bashrc
    fi
    # Instal.lem dependències del nou idioma per tal de traduir-ho tot.
    apt-get install $(check-language-support -l "$locale") -y
}
set_layout() {

    layout="$1"
    variant="$2"

    apt install xrdp -y
    if [ $# -ne 2 ]; then
        echo -e "It is necessary to pass 2 arguments, the first corresponding to the
\nlayout and the second to the variant"
        exit 1
    fi

    if [ "$layout" == "$variant" ]; then
        variant="basic"
    fi
    if [ $(( [ "$DISPLAY" ] || [ "$WAYLAND_DISPLAY" ] || [ "$MIR_SOCKET" ] && echo 1 ) ||
echo 0) -eq 1 ]; then
        if [ $(echo "$DISPLAY") = ":0" ]; then
            setxkbmap -layout $layout -variant $variant

```

```

    fi
    fi
    if [ $(grep "setxkbmap -layout $layout -variant $variant" /etc/profile | wc -l) -eq
0 ]; then
        sed -i '/^setxkbmap/d' /etc/profile
        echo "if [ \${([ "\$DISPLAY" ] || [ "\$WAYLAND_DISPLAY" ] || [ "\$MIR_SOCKET" ]
&& echo 1) || echo 0) -eq 1 ]; then
            if [ \$(echo "\$DISPLAY") = ":0" ]; then
                setxkbmap -layout $layout -variant $variant
            fi
        fi" >> /etc/profile
    fi
    if [ $(grep "setxkbmap -layout $layout -variant $variant" /etc/bash.bashrc | wc -l) -eq
0 ]; then
        sed -i '/^setxkbmap/d' /etc/bash.bashrc
        echo "if [ \${([ "\$DISPLAY" ] || [ "\$WAYLAND_DISPLAY" ] || [ "\$MIR_SOCKET" ]
&& echo 1) || echo 0) -eq 1 ]; then
            if [ \$(echo "\$DISPLAY") = ":0" ]; then
                setxkbmap -layout $layout -variant $variant
            fi
        fi" >> /etc/bash.bashrc
    fi
    if [ $(grep "setxkbmap -layout $layout -variant $variant" /root/.profile | wc -l) -eq 0
]; then
        sed -i '/^setxkbmap/d' /root/.profile
        echo "if [ \${([ "\$DISPLAY" ] || [ "\$WAYLAND_DISPLAY" ] || [ "\$MIR_SOCKET" ]
&& echo 1) || echo 0) -eq 1 ]; then
            if [ \$(echo "\$DISPLAY") = ":0" ]; then
                setxkbmap -layout $layout -variant $variant
            fi
        fi" >> /root/.profile
    fi
    if [ $(grep "setxkbmap -layout $layout -variant $variant" /root/.bashrc | wc -l) -eq 0
]; then
        sed -i '/^setxkbmap/d' /root/.bashrc
        echo "if [ \${([ "\$DISPLAY" ] || [ "\$WAYLAND_DISPLAY" ] || [ "\$MIR_SOCKET" ]
&& echo 1) || echo 0) -eq 1 ]; then
            if [ \$(echo "\$DISPLAY") = ":0" ]; then
                setxkbmap -layout $layout -variant $variant
            fi
        fi" >> /root/.bashrc
    fi
}

```

8.2.8 dhcp_script.sh

```

#!/bin/bash
scripts_path=$(cat /etc/odroid_cluster.conf | grep "^SCRIPTS_DIR=" | cut -d= -f2)
# Carreguem el script network_lib.sh com a una llibreria, per
# poder fer servir les seves funcions
source network_lib.sh
LOG_FILE=/var/log/dnsmasq.log
HOSTS_FILE=/etc/dnsmasq.d/dnsmasq_hosts.conf
host_name=$(cat /etc/odroid_cluster.conf | grep "^HOSTS_NAME=" | cut -d= -f2)
option=$1
mac=$2
ip=$3
get_time(){
    expr $(date +%s) / 60
}
# Funció per calcular els temps de sleep entre upgrade i upgrade
get_sleep_time() {
    upgrade_time=$(cat /etc/odroid_cluster.conf | grep "^UPGRADE_SLEEP=" | cut -d= -f2)
    act_time=$(get_time)
    if [ -f "${scripts_path}/last_upgrade" ]; then
        end_time=$(cat "${scripts_path}/last_upgrade")
        sleep_time=$(expr $end_time - $act_time)
        if [ $end_time -gt $act_time ]; then
            echo "$(expr $end_time + $upgrade_time)" >
"${scripts_path}/last_upgrade"
            echo "$sleep_time"
        else
            echo "$(expr $act_time + $upgrade_time)" >
"${scripts_path}/last_upgrade"
        fi
    fi
}

```

```

        echo "0"
    fi
else
    echo "$((expr $act_time + $upgrade_time))" > "${scripts_path}/last_upgrade"
    echo "0"
fi
}
# Funció per afegir una nova odroid al fitxer de hosts i configurar-la
add_odroid() {
    if [[ ! $(cat $HOSTS_FILE | grep ^dhcp-host=$mac) ]]; then
        num_line=$((expr $(cat $HOSTS_FILE | grep ^dhcp-host | wc -l) + 1))
        # Assignem una ip i hostname fix a la nova MAC
        # (Aquests canvis nomes es produeixen al reiniciar dnsmasq)
        if [[ "$num_line" -gt $(cat $HOSTS_FILE | wc -l) ]]; then
            echo "dhcp-host=$mac,$host_name$num_line,$ip" >> $HOSTS_FILE
        else
            sed -i ""$num_line""i\dhcp-
host="$mac", "$host_name""$num_line", "$ip"" $HOSTS_FILE
        fi
        # Modifiquem el nombre de odroids al fitxer slurm.conf
        sed -i 's/"$host_name"[0-9]*"/"$host_name"[1-"$num_line"]/g'
/etc/slurm-llnl/slurm.conf
        # Fem un restart del dimoni slurmctld
        systemctl restart slurmctld
        # Assignem un hostname a una ip de forma temporal.
        # (Aquests canvis es realitzen automaticament)
        echo "${ip} ${host_name}${num_line}" >> /etc/hosts.d/tmp_hosts
        # Afegim el nou slave i l'inicialitzem
        nohup "${scripts_path}/add_slave.sh" "${host_name}${num_line}"
"$get_sleep_time" >> /var/log/odroid_cluster/add_slave_"${host_name}${num_line}"".out 2>&1
&
    fi
}
# Funció per esborrar una odroid del fitxer de hosts
delete_odroid() {
    sed -i '/'$mac"/d' /var/lib/misc/dnsmasq.leases
    #sed '/^$/d' /var/lib/misc/dnsmasq.leases # Esborra línies buides
    sed -i '/'$mac"/d' $HOSTS_FILE
    #sed '/^$/d' $HOSTS_FILE # Esborra línies buides
}
# Funció per afegir una nova entrada al fitxer de log
save_log() {
    echo $(date) >> $LOG_FILE
    echo $@ >> $LOG_FILE
    echo >> $LOG_FILE
}
if [[ ! -f $HOSTS_FILE ]]; then
    touch $HOSTS_FILE
fi
save_log $@
case "$option" in
    add)
        add_odroid $@
        ;;
    old)
        add_odroid $@
        ;;
    del)
        #delete_odroid $@
        ;;
    -h|--help)
        echo "Dnsmasq sends the following parameters in this order:
\$1 = Action to take, which can be: add, old, del
\$2 = mac address
\$3 = IP address
\$4 = Assigned name
Of the above, this script only uses the first 3, to assign
the IPs to the hosts statically and give them a name."
        ;;
    *)
        echo "Incorrect option" 1>&2
        exit 1
        ;;
esac

```

8.2.9 *add_slave.sh*

```
#!/bin/bash
scripts_path="$(cat /etc/odroid_cluster.conf | grep "^SCRIPTS_DIR=" | cut -d= -f2)"
upgrade_slave="$(cat /etc/odroid_cluster.conf | grep "^UPGRADE=" | cut -d= -f2)"
# Carreguem el script network_lib.sh com a una llibreria, per
# poder fer servir les seves funcions
source network_lib.sh
# Evitem que es guardi el password al historial
export HISTIGNORE=${HISTIGNORE}:'*sudo -S:*sshpass*'
default_password=$(cat /etc/odroid_cluster.conf | grep "^DEFAULT_PASSWORD=" | cut -d= -f2)
# Agafem el nom de l'usuari no root
user_name=$(cat /etc/odroid_cluster.conf | grep "^DEFAULT_USER=" | cut -d= -f2)
# Agafem el directori home l'usuari no root
user_home=$(eval echo "~$user_name")
upgrade_time=$(cat /etc/odroid_cluster.conf | grep "^UPGRADE_SLEEP=" | cut -d= -f2)
language=$(cat /etc/odroid_cluster.conf | grep "^SYS_LANGUAGE=" | cut -d= -f2)
layout=$(cat /etc/odroid_cluster.conf | grep "^LAYOUT=" | cut -d= -f2)
variant=$(cat /etc/odroid_cluster.conf | grep "^VARIANT=" | cut -d= -f2)
timezone=$(cat /etc/odroid_cluster.conf | grep "^SYS_TIMEZONE=" | cut -d= -f2)
default_host=$(cat /etc/odroid_cluster.conf | grep "^HOSTS_NAME=" | cut -d= -f2)
locale="$language\;$layout\;$variant\;$timezone"
KEY_FILE="{user_home}/.ssh/id_rsa.pub"
KNOWN_HOSTS="{user_home}/.ssh/known_hosts"
host="$1" # $1 la ip del slave
passphrase="$2" # $4 passphrase
# Comprovem que es passi com a minim 1 parametre
if [ $# -lt 1 ]; then
    echo -e "Error, at least you have to enter 1 parameter, for more information \n\t
add_slave -h"
    exit 1
fi
# Afegim el fingerprint al fitxer de hosts coneguts
su $user_name -c "echo \"$(ssh-keyscan -H $host)\ " >> $KNOWN_HOSTS"
# Copiem la clau publica al slave
su $user_name -c "sshpass -p $default_password ssh-copy-id -i $KEY_FILE $user_name@$host"
# Copiem el script locale.sh dependencia de init_slave.sh
su $user_name -c "scp ${scripts_path}/locale.sh ${user_name}@${host}:Documents"
# Copiem el script de inicialització al slave
su $user_name -c "scp ${scripts_path}/init_slave.sh ${user_name}@${host}:Documents"
# Agafem la IP de la xarxa interna
interface="$(cat /etc/dnsmasq.conf | grep interface= | cut -d= -f2)"
master_ip="$(get_ip_of_nic $interface)"
# Executem el script de inicialització al slave
su $user_name -c "ssh -t ${user_name}@${host} \"echo ${default_password} | sudo -S
~/Documents/init_slave.sh ${master_ip} ${upgrade_slave} ${upgrade_time} ${locale} \" >>
/var/log/odroid_cluster/init_slave_${host}.out 2>&1"
```

8.2.10 *init_slave.sh*

```
#!/bin/bash
# Carreguem el script locale.sh com a una llibreria, per
# poder fer servir les seves funcions
source $(echo "`dirname \"$0\"`"/locale.sh
SLURM_ETC=/etc/slurm-llnl
add_resolvconf() {
    dns_ip="$1"
    # Instal·lem el dimoni resolvconf
    apt-get install resolvconf -y
    # Habilitem i reiniciem el dimoni de resolvconf
    systemctl enable resolvconf
    systemctl start resolvconf
    # Copiem el contingut de original a tail, per a que renovi el contingut
    echo "nameserver ${dns_ip}" > /etc/resolvconf/resolv.conf.d/head
    # Actualitzem els DNS
    resolvconf --enable-updates
    resolvconf -u
}
add_slurm() {
    master_ip=$1
    # Instal·lem slurm-wlm
```

```

    apt install slurm-wlm -y
    if [ $(cat /etc/fstab | grep "${SLURM_ETC}" | wc -l) -eq 0 ]; then
        echo "${master_ip}:${SLURM_ETC} ${SLURM_ETC} nfs rw,auto,_netdev 0 0" >>
/etc/fstab
    fi
    mount ${SLURM_ETC}
    mkdir /var/spool/slurmd
    chown slurm: /var/spool/slurmd
    systemctl enable --now slurmd
}
add_slurm_watcher() {
    echo "[Unit]
Description=Restart slurmd if slurm.conf is modified.
After=network.target
[Service]
Type=oneshot
ExecStart=$(which systemctl) restart slurmd.service
[Install]
WantedBy=multi-user.target" > /etc/systemd/system/slurm_watcher.service
    echo "[Path]
PathModified=${SLURM_ETC}/slurm.conf
[Install]
WantedBy=multi-user.target" > /etc/systemd/system/slurm_watcher.path
    systemctl enable --now slurm_watcher.{path,service}
}
if [ $# -ne 4 ]; then
    echo "Error, you must enter 5 parameters, the first corresponding to the IP or host
name
of the master, the second an integer value between 1 and 0 to indicate whether the slave is
updated or not and the third an integer value corresponding to the waiting time in minutes
between the upgrade of one node and the next."
    exit 1
fi
if [ "$2" -ne 1 ] && [ "$2" -ne 0 ]; then
    echo "The second parameter must be an integer value between 1 and 0"
    exit 1
fi
master_ip="$1" # $1 ip del master a la lan odroid
upgrade="$2"
sleep_time="$3"
locale="$4"
language=$(echo "$locale" | cut -d ";" -f 1)
layout=$(echo "$locale" | cut -d ";" -f 2)
variant=$(echo "$locale" | cut -d ";" -f 3)
timezone=$(echo "$locale" | cut -d ";" -f 4)
# Deshabilitem Unattended-Upgrade
systemctl disable unattended-upgrades
systemctl stop unattended-upgrades
apt remove unattended-upgrades -y
nic=$(echo $(sed '1d;2d' /proc/net/dev | grep -v 'lo' | cut -d: -f1))
if [ -z "$nic" ]; then
    echo "Error, no NIC found"
    exit 1
fi
# Bucle de espera, per assegurar-nos de que la resoluci3 de noms est3 funcionant
correctament
while [[ $(ping google.com -I "$nic" -w2 2> /dev/null | grep "received" | cut -d " " -f4) -
eq 0 ]]; do
    sleep 2
done
# Fixem com a DNS el master
rm /etc/resolv.conf
echo "nameserver ${master_ip}" > /etc/resolv.conf
chattr +i /etc/resolv.conf
# Configurem les interfícies de xarxa
echo "auto lo
iface lo inet loopback
auto ${nic}
iface ${nic} inet dhcp" > /etc/network/interfaces
# Afegim aquest petit sctipt per a que actualitci el hostname amb el dhcp
echo "#!/bin/bash
hostnamectl set-hostname --static \${new_host_name}" > /etc/dhcp/dhclient-exit-
hooks.d/hostname
chmod a+r /etc/dhcp/dhclient-exit-hooks.d/hostname
dhclient -v
# Fiquem a zona horaria i actualitzem l'hora

```

```

timedatectl set-timezone "$timezone"
apt install chrony -y
systemctl enable --now chronyd
# Solucionem error de claus amb l'update
apt-key adv -v --keyserver keyserver.ubuntu.com --recv-keys 5360FB9DAB19BAC9
# Actualitzem
apt-get update -y
# Instal·lem sysstat per a al software de monitorització de Joan Jara
apt-get install sysstat -y
# Desactivem autenticació mitjançant usuari root
sed -i 's/PermitRootLogin yes.*/PermitRootLogin no/' /etc/ssh/sshd_config
systemctl restart sshd
# Esborrem software innecessari
apt-get remove --purge libreoffice* thunderbird pacman transmission* -y
apt autoremove -y
apt autoclean -y
# Instal·lem el client NFS
apt-get install nfs-common -y
# Afegim NFS /home a fstab i el muntem
echo "${master_ip}:/home /home nfs rw,auto,_netdev 0 0" >> /etc/fstab
mount -a || echo "Error: Check the /etc/fstab file, probably the shared directory could
not be mounted using NFS (Network File System), do not restart
$(hostname) before solving this problem."
# Instal·lem munge
apt-get install munge -y
systemctl enable --now munge
# Copiem la clau guardada a /home i reiniciem munge
dd if=/home/munge.key of=/etc/munge/munge.key
systemctl restart munge
add_slurm "${master_ip}"
add_slurm_watcher
apt-get install mpich -y
echo "I am $(hostname) I have already installed and configured everything." >>
~/.slave_responses
# Modifiquem l'idioma i el layout del teclat
set_language "$language"
set_layout "$layout" "$variant"
if [ $upgrade -eq 1 ]; then
    sleep "$sleep_time"m && apt-get upgrade -y &> /var/log/upgrade_$(hostname).log
fi
# Esborrem el propi script
rm -- "$0"

```

8.2.11 check_gui_dependencies.sh

```

#!/bin/bash
system=$(cat dependencies | grep ^system | cut -d: -f2-)
python=$(cat dependencies | grep ^python3 | cut -d: -f2-)
IFS=: read -a sys_array <<< "$system"
IFS=: read -a py_array <<< "$python"
for i in "${sys_array[@]"; do
    if [ $(dpkg -l $i &>/dev/null ; echo $? ) -eq 1 ]; then
        exit 1
    fi
done
pip3_list=$(pip3 list)
for i in "${py_array[@]"; do
    if [ $(echo "$pip3_list" | grep "\<$i\>" | wc -l) -eq 0 ]; then
        exit 1
    fi
done
exit 0

```

8.2.12 gui_dependencies.sh

```

#!/bin/bash
system=$(cat dependencies | grep ^system | cut -d: -f2-)
python=$(cat dependencies | grep ^python3 | cut -d: -f2-)

```



```
apt install $(echo ${system//:/ }) -y
pip3 install $(echo ${python//:/ })
```

8.2.13 *installer_gui.pyw*

```
#!/usr/bin/python3
# -*- coding: utf-8 -*-
import subprocess
from pathlib import Path
import time
import os
uid=os.getuid()
ret=subprocess.run(str(Path.cwd())+"/check_gui_dependencies.sh", shell=True).returncode
if(int(ret) == 1):
    if(uid == 0):
        subprocess.call(str(Path.cwd())+"/gui_dependencies.sh", shell=True)
    else:
        subprocess.call("sudo -k "+str(Path.cwd())+"/gui_dependencies.sh",
shell=True)
from tkinter import ttk
from tkinter import *
from PIL import ImageTk,Image
import time
from fontTools.ttLib import TTFont
from tkinter import messagebox
import tkinter.font as tkFont
import re
from icu import Locale
if(uid > 0):
    font = TTFont('optima-roman.ttf')
    font.save(str(Path.home())+"/.local/share/fonts/optima-roman.ttf")
else:
    font = TTFont('optima-roman.ttf')
    font.save("/usr/share/fonts/optima-roman.ttf")
OPTIONS_FILE="odroid_cluster.conf"
MAIN_COLOR='#90292A'
BACKGROUND_COLOR='#fafafa'
FONT="Optima"
TEXT_FONT=0
FONT_SIZE=15
theme= {
    "TNotebook.Tab": {"configure": {"padding": [80, 10],
                                "background": "#231f20",
                                "foreground": "white",
                                "font" : (FONT, FONT_SIZE)
                                },
                    "map": {"background": [("selected", MAIN_COLOR),
                                        ("active",
                                        "#AA554F")]
                    }
    },
    "TButton": {"configure": {
                                "padding": [30, 10],
                                "background":
                                MAIN_COLOR,
                                "foreground":
                                "white",
                                "font" : (FONT,
                                FONT_SIZE)
                                },
                    "map": {"background": [("selected",
                                MAIN_COLOR),
                                ("active", "#AA554F")]
                    }
    },
    "TRadiobutton": {"configure": {
                                "padding": [0,5],
                                "background":
                                BACKGROUND_COLOR,
                                "font" :
                                (TEXT_FONT, FONT_SIZE)
                                },
                    "map": {"background": [("selected",
                                BACKGROUND_COLOR),
```

```

("active", BACKGROUND_COLOR)]
    },
    "TCheckbutton": {"configure": {
        "padding": [0,5],
        "background":
BACKGROUND_COLOR,
        "font" :
(TEXT_FONT, FONT_SIZE)
    },
    "map": {"background": [("selected",
BACKGROUND_COLOR),
("active", BACKGROUND_COLOR)]
    }
    "TEntry": {"configure": {
        "padding": [0,5],
        "background":
BACKGROUND_COLOR,
        "font" :
(TEXT_FONT, FONT_SIZE)
    },
    "TLabel": {"configure": {
        "padding": [0,5],
        "background":
BACKGROUND_COLOR,
        "font" :
(TEXT_FONT, FONT_SIZE)
    },
    "TCombobox": {"configure": {
        "padding": [0, 5],
        "background":
BACKGROUND_COLOR,
        "font" : (FONT,
FONT_SIZE),
        'selectbackground': 0,
        'selectforeground': 'black'
    },
    "map": {"background": [("selected",
BACKGROUND_COLOR),
("active", BACKGROUND_COLOR),
('readonly', BACKGROUND_COLOR)],
        "fieldbackground" :
[('readonly', 'white')]
    }
    "TFrame": {"configure": {
        "background": BACKGROUND_COLOR
    }
    }
}

class Variant:
    def __init__(self, name, code):
        self.name=name
        self.code=code

    def get_name(self):
        return self.name

    def get_code(self):
        return self.code

class Layout:
    def __init__(self, name, code):
        self.name=name
        self.code=code
        self.variants=[]

    def add_variant(self, variant):
        self.variants.append(variant)

    def get_name(self):

```

```

        return self.name

def get_code(self):
    return self.code

def get_variant(self, index):
    try:
        return self.variants[index]
    except(IndexError):
        return None

def get_num_variants(self):
    return len(self.variants)

def get_variants(self):
    return self.variants

def get_variants_name(self):
    names=[]
    for i in self.variants:
        names.append(i.get_name())
    return names

def get_variants_code(self):
    codes=[]
    for i in self.variants:
        codes.append(i.get_code())
    return codes

def get_timezones():
    global timezones
    p = subprocess.run(['timedatectl', 'list-timezones'], universal_newlines=True,
stdout=subprocess.PIPE, stderr=subprocess.PIPE)
    #p.stderr
    timezones = list(p.stdout.split("\n"))

def get_languages():
    global icu
    global languages
    icu = []
    languages = []
    #cat /usr/share/i18n/SUPPORTED | grep UTF-8 | grep -v @ | awk '{print $1}' | cut -d.
-f1
    p = subprocess.run(['cat', '/usr/share/i18n/SUPPORTED'], universal_newlines=True,
stdout=subprocess.PIPE, stderr=subprocess.PIPE)
    aux = list(p.stdout.split("\n"))
    # Seleccionem els country code
    for lang in aux:
        if(re.search(r'UTF-8', lang)):
            if(re.search(r'@', lang) is None):
                lang=lang.split()[0]
                icu.append(lang.split('.')[0])
                languages.append(Locale(lang).getDisplayname())

def read_layout_file(file, path):
    f = open(path+"/"+file, "r")

    code=None
    name=None
    layout_name=None

    for line in f.readlines():
        if(re.search(r'xkb_symbols', line)):
            code = re.split("'", line)[1].rstrip()
        elif(re.search(r'name\[', line)):
            name = re.split("'", line)[1].rstrip()
            if(code != "basic" and name == layout_name):
                name = name + " " + code
        if(code == "basic" and name != None):
            layout_name=name

    if(layout_name != None and code != None and name != None):
        if(code == "basic"):
            layouts[name]=Layout(name, file)
            layouts[layout_name].add_variant(Variant(name, file))
        else:
            layouts[layout_name].add_variant(Variant(name, code))
    code = None

```

```

        name = None

def get_layouts():
    global layouts

    layouts = {}
    files = []
    path = '/usr/share/X11/xkb/symbols'
    p = subprocess.run(['ls', '-p', path], universal_newlines=True,
stdout=subprocess.PIPE, stderr=subprocess.PIPE)
    for file in (p.stdout.split("\n")):
        if(re.search(r'/', file) is None and file != ''):
            files.append(file)

    for i in files:
        read_layout_file(i,path)
    layouts.pop('Empty', None)
def get_pos_list(element, elements):
    line = 0
    counter=0
    for i in elements:
        if(element == i):
            line = counter
            counter += 1
    return(line)

def start(window, password):
    res=check_password(password)
    if(res == 0):
        window.destroy()
        subprocess.call("x-terminal-emulator -e 'echo \""+password+"\" | sudo -Sk
"+str(Path().absolute()+"/init_master.sh'", shell=True)
    else:
        password=""
        window.entry.delete(0,END)
        window.geometry("400x280")
        center_window(window)
        window.error_text.set("Wrong password!")
def check_password(password):
    res=subprocess.run("echo \""+password+"\" | sudo -Sk echo \"correct password\" 2>
/dev/null", shell=True).returncode
    return(res)
def ask_password(window):
    password = StringVar()
    pass_window = Toplevel(window)
    pass_window.error_text = StringVar()

    # Modifiquem el color de fons, titol i tamany
    pass_window.configure(background=BACKGROUND_COLOR)
    pass_window.title("")
    pass_window.geometry("400x250")
    center_window(pass_window)

    ttk.Label(pass_window, text="Enter your password").pack(pady=20)
    pass_window.entry=ttk.Entry(pass_window, textvariable=password, show="*")
    pass_window.entry.pack(pady=20)
    ttk.Button(pass_window, text="Start", command= lambda: start(pass_window,
password.get())).pack(pady=20, side=BOTTOM)
    ttk.Label(pass_window, textvariable=pass_window.error_text, foreground="red").pack()

def read_option(option):
    f = open(OPTIONS_FILE, "r")
    found=None

    for line in f:
        splited=line.split("=")
        if(splited[0]==option):
            found=splited[1]
    f.close()

    return(found.rstrip())
def write_option(option, value):
    f = open(OPTIONS_FILE, "r")
    found=False

```

```

lines = f.readlines()
num_lines=len(lines)

i=0
while(found==False and i<num_lines):
    if(lines[i].split("=")[0]==option):
        found=True
        lines[i]=option+"="+value
    i+=1
f.close()

f = open(OPTIONS_FILE, "w")

for line in lines:
    f.write(line.rstrip()+"\n")
f.close()

return(found)
def iplist_to_ipstring(list1):
    ip= ""

    for i in list1:
        ip+=i.get("1.0",END).rstrip()
        ip+="."

    return(ip[:-1])
def check_ip(list1, title_error):

    for i in list1:
        i=i.get("1.0",END)
        try:
            int(i)
        except ValueError:
            messagebox.showerror(message='The IP can only contain digits',
title=title_error)
            return False
        if(int(i) < 0 or int(i) > 255):
            messagebox.showerror(message='The digits of the IP can only be
between 0 and 255', title=title_error)
            return False
    return True

def center_window(window):
    # Agafem l'amplada i alçada de la pantalla
    window_width = window.wininfo_reqwidth()
    window_height = window.wininfo_reqheight()

    # Calculem la posició central agafant lamplada i alçada de la pantalla
    position_x = int(window.wininfo_screenwidth()/2 - window_width/2)
    position_y = int(window.wininfo_screenheight()/2 - window_height/2)

    # Canviem la posició de la finestra
    window.geometry("+%d+%d" % (position_x, position_y))
def add_image(container, scale, path):
    image = Image.open(path)
    width, height = image.size
    image=image.resize((int(width*scale), int(height*scale)))
    photo = ImageTk.PhotoImage(image)
    layer = Label(container, image=photo , background=MAIN_COLOR)
    layer.image = photo
    return(layer)
def load_screen(window):
    # Treiem la barra superior
    window.overrideredirect(1)
    screen_width=window.wininfo_screenwidth()
    screen_height=window.wininfo_screenheight()
    window.configure(bg=MAIN_COLOR)
    f1=Frame(window, background=MAIN_COLOR)
    l1=add_image(f1,0.5,"odroid_cluster.png")
    l3=add_image(f1,0.2,"urv.png")
    var = StringVar()
    l2 = Label(f1, textvariable = var, background=MAIN_COLOR, fg="white", font=(FONT,30))
    var.set("Odroid Cluster")
    l1.pack(padx=40, pady=10)
    l2.pack(padx=10, pady=20)

```

```

l3.pack(side=RIGHT)
f1.pack(padx=70, pady=30)
# Apparently a common hack to get the window size. Temporarily hide the
# window to avoid update_idletasks() drawing the window in the wrong
# position.
root.withdraw()
root.update_idletasks() # Update "requested size" from geometry manager
center_window(window)
# This seems to draw the window frame immediately, so only call deiconify()
# after setting correct window position
window.deiconify()
def add_menu(window):
    window.notebook = ttk.Notebook(window)
    window.f1=Frame(window, background=BACKGROUND_COLOR)
    window.f2=Frame(window, background=BACKGROUND_COLOR)
    window.f3=Frame(window, background=BACKGROUND_COLOR)
    # Añadirlas al panel con su respectivo texto.
    window.notebook.add(window.f1, text="Install")
    window.notebook.add(window.f2, text="Advanced")
    window.notebook.add(window.f3, text="Slurm")
def add_simple_dropdown(window, elements, variable, side=RIGHT):
    position = get_pos_list(variable.get(), elements)
    combo = ttk.Combobox(window, state="readonly", textvariable=variable,
values=elements)
    combo.config(font=(TEXT_FONT, FONT_SIZE))
    if(position >= 0):
        combo.current(position)
    combo.pack(fill=X, side=side)

    return(combo)
def add_dropdown(window, elements, variable, text=None):

    frame = ttk.Frame(window)

    combo = add_simple_dropdown(frame, elements, variable)

    label=ttk.Label(frame, text=text)

    label.pack( fill=X, side=LEFT)
    frame.pack(expand=True, fill=BOTH, side=TOP)

    return(combo)

def add_ip_dropdown(window, elements, variable, combo2=None, variable2=None):
    position = get_pos_list(variable.get(), elements)
    combo = ttk.Combobox(window, state="readonly", textvariable=variable,
values=elements, width=3)

    combo.config(font=(TEXT_FONT, FONT_SIZE))
    if(position >= 0):
        combo.current(position)
    combo.pack(side=LEFT)

    return(combo)
def add_label(frame, text):
    label = ttk.Label(frame, text =
text, background=BACKGROUND_COLOR, font=(TEXT_FONT, FONT_SIZE))
    label.pack(side=LEFT)
    return(label)

def add_entry(window, default_text, label_text):
    frame = ttk.Frame(window)

    add_label(frame, label_text)
    entry = ttk.Entry(frame, font=(TEXT_FONT, FONT_SIZE))
    entry.insert(INSERT, default_text)
    entry.pack(side=RIGHT)

    frame.pack(expand=True, fill=X)

    return(entry)
def add_radiobutton(window, text, var, val, command):
    radio_button = ttk.Radiobutton(window, text=text, variable=var, value=val,
command=command)
    return(radio_button)

```

```

def add_radiobutton_group(window, label_text, var, text_list, options_list, commands=None):
    private_ip_frame = ttk.Frame(window)

    label=add_label(private_ip_frame, label_text)

    i2=0
    for i in text_list:
        add_radiobutton(private_ip_frame, i, var, options_list[i2],
command=commands[i2]).pack(side=LEFT, expand=True,)
        i2+=1

    private_ip_frame.pack(expand=True, fill=X)

def add_checkbutton(window, text, var):

    check_button = ttk.Checkbutton(window, text=text, variable=var, compound=LEFT)
    check_button.pack(side=LEFT)

    return(check_button)

def add_ip(window, default_ip, text):
    ip = []

    frame = ttk.Frame(window)
    subframe = ttk.Frame(frame)

    label = add_label(frame, text)

    for i in range(4):
        ip.append(Text(subframe, height=1, width=3,font=(TEXT_FONT,FONT_SIZE)))

    c=len(ip)-1
    for i in reversed(ip):
        i.pack(side=RIGHT)
        if(c > 0):
            Label(subframe, text = " .
",background=BACKGROUND_COLOR,font=(TEXT_FONT,FONT_SIZE)).pack(side=RIGHT)
            i.insert(INSERT,default_ip[c])
            c-=1
        subframe.pack(expand=True, fill=X)
        frame.pack(expand=True, fill=X)
    return(ip)

def int_to_byte(number):
    byte=str("{0:8b}".format(number))

    return(byte)

def cidr_to_mask(cidr):
    conv_mask=""

    i2=0

    for i in range(0,cidr):
        if(i2%8 == 0 and i2 !=0):
            conv_mask = conv_mask + ":"
            conv_mask = conv_mask + "1"
            i2+=1

    for i in range(cidr,32):
        if(i2%8 == 0 and i2 !=0):
            conv_mask = conv_mask + ":"
            conv_mask = conv_mask + "0"
            i2+=1

    int_mask=conv_mask.split(":")

    conv_mask=str(int(int_mask[0], 2))
    conv_mask = conv_mask + ":"
    conv_mask=conv_mask + str(int(int_mask[1], 2))
    conv_mask = conv_mask + ":"
    conv_mask=conv_mask + str(int(int_mask[2], 2))
    conv_mask = conv_mask + ":"
    conv_mask=conv_mask + str(int(int_mask[3], 2))

    return(conv_mask)

def mask_to_cidr(my_mask):
    bit_mask=""

```

```

for i in my_mask.split(":"):
    bit_mask+=str('{:08b}'.format(int(i)))

count = 0
for i in bit_mask:
    if(i == '1'):
        count = count + 1
return(count)

""" Funció per a resetejar els valors de les variables que formen el rang de IP privada A,
a més
    modifica els valors per defecte dels dos primers combobox que formen les dues IP,
també modifica
    els parametres de la funció que es crida al modificar el combobox del segon nombre
de la ip """
def set_ip_a():
    mask_values = []
    ip.ip_num1.set(10)
    ip.ip_num2.set(0)
    ip.ip_num3.set(0)
    ip.ip_num4.set(1)
    ip.combo1['values'] = "10"
    ip.combo2['values'] = number_list
    mask.min_value=8
    mask.set(8)
    for i in range(mask.min_value, 32):
        mask_values.append(i)
    mask.combo_mask['values'] = mask_values
""" Funció per a resetejar els valors de les variables que formen el rang de IP privada B,
a més
    modifica els valors per defecte dels dos primers combobox que formen les dues IP,
també modifica
    els parametres de la funció que es crida al modificar el combobox del segon nombre
de la ip """
def set_ip_b():
    elements=[]

    for i in range(16,32):
        elements.append(i)

    mask_values = []
    ip.ip_num1.set(172)
    ip.ip_num2.set(16)
    ip.ip_num3.set(0)
    ip.ip_num4.set(1)
    ip.combo1['values'] = "172"
    ip.combo2['values'] = elements
    mask.min_value=12
    mask.set(16)
    for i in range(mask.min_value, 32):
        mask_values.append(i)
    mask.combo_mask['values'] = mask_values

""" Funció per a resetejar els valors de les variables que formen el rang de IP privada C,
a més
    modifica els valors per defecte dels dos primers combobox que formen les dues IP """
def set_ip_c():
    mask_values = []
    ip.ip_num1.set(192)
    ip.ip_num2.set(168)
    ip.ip_num3.set(0)
    ip.ip_num4.set(1)
    ip.combo1['values'] = "192"
    ip.combo2['values'] = "168"

    mask.set(24)
    mask.min_value=16
    for i in range(mask.min_value, 32):
        mask_values.append(i)
    mask.combo_mask['values'] = mask_values
""" Funció per afegir un 8 combobox per poder afegir el rang de ip
privada """
def add_private_ip(window, frame):
    # Declarem les variables necessaries com a globals
    global ip

```



```

global number_list
list_ip_b=[]

ip = ttk.Frame(window);

ip.ip_num1 = IntVar()
ip.ip_num2 = IntVar()
ip.ip_num3 = IntVar()
ip.ip_num4 = IntVar()

number_list=[]
number_list_end=[]

for i in range(0,256):
    number_list.append(str(i))

for i in range(1,255):
    number_list_end.append(str(i))

for i in range(16,32):
    list_ip_b.append(i)

subframe = ttk.Frame(frame)

add_mask(window, subframe)

add_label(frame, "IP: ")

if(window.f2.radio1.get() == 'B'):
    ip.combo1=add_ip_dropdown(subframe, ["172"], ip.ip_num1)
elif(window.f2.radio1.get() == 'C'):
    ip.combo1=add_ip_dropdown(subframe, ["192"], ip.ip_num1)
else:
    ip.combo1=add_ip_dropdown(subframe, ["10"], ip.ip_num1)
add_label(subframe, ".")
if(window.f2.radio1.get() == 'B'):
    ip.combo2=add_ip_dropdown(subframe, list_ip_b, ip.ip_num2)
elif(window.f2.radio1.get() == 'C'):
    ip.combo2=add_ip_dropdown(subframe, ["168"], ip.ip_num2)
else:
    ip.combo2=add_ip_dropdown(subframe, number_list, ip.ip_num2)
add_label(subframe, ".")
ip.combo3=add_ip_dropdown(subframe, number_list, ip.ip_num3)
add_label(subframe, ".")
ip.combo4=add_ip_dropdown(subframe, number_list_end, ip.ip_num4)

splited_ip=read_option("IP").split(".")

ip.ip_num1.set(splited_ip[0])
ip.ip_num2.set(splited_ip[1])
ip.ip_num3.set(splited_ip[2])
ip.ip_num4.set(splited_ip[3])
subframe.pack(expand=True, fill=BOTH)

mask.set(int(mask_to_cidr(read_option("MASK"))))
def add_mask(window, frame):
    global mask

    mask_values=[]
    mask = IntVar()
    #mask.set(int(mask_to_cidr(read_option("MASK"))))

    if(window.f2.radio1.get() == 'C'):
        mask.min_value=16

    elif(window.f2.radio1.get() == 'B'):
        mask.min_value=12

    elif(window.f2.radio1.get() == 'A'):
        mask.min_value=8

    for i in range(mask.min_value, 32):
        mask_values.append(i)

```

```

mask.combo_mask = add_ip_dropdown(frame, mask_values, mask)
add_label(frame, "Mask: ").pack(side=RIGHT)
mask.combo_mask.pack(side=RIGHT)
def get_list(dict):
    list = []
    for key in dict.keys():
        list.append(key)
    return list
def refresh_variants(combo, combo2, var, var2):
    combo2['values'] = sorted(layouts[var.get()].get_variants_name())
    var2.set(var.get())

def key_of_value(dict, value2):
    for key, value in dict.items():
        if(value.get_code() == value2):
            return key
    return None
def add_listbox(window, elements):
    listbox = Listbox(window, font=(TEXT_FONT, FONT_SIZE))
    scrollbar = Scrollbar(window)
    scrollbar_x = Scrollbar(window, orient=HORIZONTAL)
    for item in elements:
        listbox.insert(END, item)

    listbox.config(yscrollcommand = scrollbar.set)
    listbox.config(xscrollcommand = scrollbar_x.set)
    scrollbar.config(command = listbox.yview)
    scrollbar_x.config(command = listbox.xview)
    scrollbar_x.pack(side = BOTTOM, fill = X)
    listbox.pack(expand=True, side = LEFT, fill = BOTH)
    scrollbar.pack(side = RIGHT, fill = BOTH)

    return(listbox)

def refresh_variants2(listb, listb2, var, var2):
    try:
        var.set(listb.get(listb.curselection()))
        listb2.delete(0,END)
        for item in sorted(layouts[var.get()].get_variants_name()):
            listb2.insert(END, item)
        var2.set(var.get())
    except TclError:
        None

def set_varaint(listb, var):
    try:
        var.set(listb.get(listb.curselection()))
    except TclError:
        None

def add_layouts(window):
    global layout
    global variant

    layout=StringVar()
    variant=StringVar()
    layout.set(key_of_value(layouts, read_option("LAYOUT")))

    for i in layouts[layout.get()].get_variants():
        if(read_option("VARIANT") == i.get_code()):
            variant.set(i.get_name())

    sorted_layouts=sorted(layouts)
    sorted_variants=sorted(layouts[layout.get()].get_variants_name())

    frame = ttk.Frame(window)
    frame_label = ttk.Frame(window)

    # Afegim una etiqueta, per indicar que 'es
    add_label(frame_label, "Select Keyboard layout:").pack(side=LEFT, anchor=S)
    frame_label.pack(expand=True, fill=BOTH)

    subframe1 = ttk.Frame(frame)
    subframe2 = ttk.Frame(frame)
    # Frame per fer un espai intermig
    padding = ttk.Frame(frame)

```

```

# Creem els dos listbox
listbox1=add_listbox(subframe1, sorted_layouts)
listbox2=add_listbox(subframe2, sorted_variants)
# Seleccionem el item del listbox segons la lectura del fitxer
listbox1.selection_set(get_pos_list(layout.get(),sorted_layouts))
listbox2.selection_set(get_pos_list(variant.get(),sorted_variants))

# Fem pack dels subframes
subframe1.pack(expand=True,side=LEFT, fill=X)
padding.pack(expand=True, side=LEFT)
subframe2.pack(expand=True,side=RIGHT, fill=X)

# Fem que el scrollbar es mogui als elements seleccionats
listbox1.see(get_pos_list(layout.get(),sorted_layouts))
listbox2.see(get_pos_list(variant.get(),sorted_variants))

# Afegim les funcions que s'executaran al seleccionar els listbox
listbox1.bind('<<ListboxSelect>>', lambda lb1=listbox1, lb2=listbox2, var=layout,
var2=variant : refresh_variants2(listbox1, listbox2, var, var2))
listbox2.bind('<<ListboxSelect>>', lambda lb=listbox2, var=variant :
set_varaint(listbox2, var))

frame.pack(expand=True, fill=BOTH, anchor=N)

def add_content_install(window):
padding_bottom = ttk.Frame(window.f1)
padding_up = ttk.Frame(window.f1)
content = ttk.Frame(window.f1)
padding_left = ttk.Frame(window.f1)
padding_right = ttk.Frame(window.f1)
padding_right2 = ttk.Frame(window.f1)

window.f1.timezone = StringVar()
window.f1.timezone.set(read_option("SYS_TIMEZONE"))
window.f1.locale = StringVar()
window.f1.locale.set(languages[get_pos_list(read_option("SYS_LANGUAGE"),icu)])

add_dropdown(content, timezones, window.f1.timezone, "Time zone:")
add_dropdown(content, languages, window.f1.locale, "Language:")

add_layouts(content)

button=ttk.Button(padding_bottom,text='INSTALL NOW', command = lambda:
start_installation(window))

padding_bottom.pack(fill=BOTH, side=BOTTOM)
padding_left.pack(expand=True, side=LEFT)
content.pack(expand=True, fill=BOTH, side=LEFT)
padding_right.pack(expand=True, fill=BOTH, side=RIGHT)
#padding_right2.pack(expand=True, fill=BOTH, side=RIGHT)

button.pack(expand=True, pady=20)

def add_content_advanced(window):
window.f2.radio1 = StringVar()
window.f2.check1 = IntVar()
window.f2.radio1.set(read_option("IP_CLASS"))
window.f2.check1.set(read_option("UPGRADE"))
text_password = Text()

window.f2.upgrade_time = StringVar()
padding_bottom = ttk.Frame(window.f2)
padding_up = ttk.Frame(window.f2)
content = ttk.Frame(window.f2)
padding_left = ttk.Frame(window.f2)
padding_right = ttk.Frame(window.f2)
padding_right2 = ttk.Frame(window.f2)
window.f2.upgrade_time.set(read_option("UPGRADE_SLEEP"))

```

```

window.f2.text_name = add_entry(content, read_option("DEFAULT_USER"), "Default
odroid user:")
window.f2.text_hostname = add_entry(content, read_option("HOSTS_NAME"), "Default
odroid hostname:")
text_password = add_entry(content, read_option("DEFAULT_PASSWORD"), "Default odroid
password:")
window.f2.text_password2 = text_password

splited_dns1=read_option("EXTERNALDNS1").split(".")
window.f2.dns1=add_ip(content,
[splited_dns1[0],splited_dns1[1],splited_dns1[2],splited_dns1[3]], "Upstream DNS server
1:")

splited_dns2=read_option("EXTERNALDNS2").split(".")
window.f2.dns2=add_ip(content,
[splited_dns2[0],splited_dns2[1],splited_dns2[2],splited_dns2[3]], "Upstream DNS server
2:")

window.f2.scripts_dir=add_entry(content,read_option("SCRIPTS_DIR"),"Scripts
directory")

frame = ttk.Frame(content)
subframe1 = ttk.Frame(frame)
subframe2 = ttk.Frame(frame)
add_checkbutton(subframe1, "Upgrade cluster",window.f2.check1)
window.f2.upgrade_entry=add_entry(subframe2, window.f2.upgrade_time.get(),"Upgrade
time: ")
subframe1.pack(side=LEFT)
subframe2.pack(side=RIGHT)
frame.pack(expand=True, fill=BOTH)

add_radiobutton_group(content,"Private IP:",window.f2.radiol, ["Class A", "Class B",
"Class C"], ['A', 'B', 'C'], [lambda :set_ip_a(), lambda :set_ip_b(), lambda :set_ip_c()])

add_private_ip(window,content)

padding_bottom.pack(fill=BOTH, side=BOTTOM)
padding_left.pack(expand=True, side=LEFT)
content.pack(expand=True, fill=BOTH, side=LEFT)
padding_right.pack(expand=True, fill=BOTH, side=RIGHT)
#padding_right2.pack(expand=True, fill=BOTH, side=RIGHT)

def add_content_slurm(window):
padding_bottom = ttk.Frame(window.f3)
padding_up = ttk.Frame(window.f3)
content = ttk.Frame(window.f3)
padding_left = ttk.Frame(window.f3)
padding_right = ttk.Frame(window.f3)
padding_right2 = ttk.Frame(window.f3)

window.f3.scheduler = StringVar()
window.f3.sched_type = IntVar()
window.f3.tracking = StringVar()
window.f3.track_type = IntVar()
window.f3.scheduler.set(read_option("SCHEDULER"))
window.f3.tracking.set(read_option("TRACKING"))

if(window.f3.scheduler.get() == "sched/builtin"):
window.f3.sched_type.set(0)
elif(window.f3.scheduler.get() == "sched/backfill"):
window.f3.sched_type.set(1)

if(window.f3.tracking.get() == "proctrack/cgroup"):
window.f3.track_type.set(0)
elif(window.f3.tracking.get() == "proctrack/pgid"):
window.f3.track_type.set(1)
elif(window.f3.tracking.get() == "proctrack/linuxproc"):
window.f3.track_type.set(2)

window.f3.max_time = add_entry(content, read_option("MAX_TIME"), "Maximum time limit
of jobs in minutes or INFINITE:")

add_radiobutton_group(content,"Scheduler type:",window.f3.sched_type, ["First-In
First-Out (FIFO)", "FIFO with backfill"], [0, 1], [None, None])

```

```

        add_radiobutton_group(content, "Process tracking type:", window.f3.track_type, ["Use
Linux cgroups", "Use Unix process group ID", "Use parent process ID"], [0, 1, 2], [None,
None, None])

padding_bottom.pack(fill=BOTH, side=BOTTOM)
padding_left.pack(expand=True, side=LEFT)
content.pack(expand=True, fill=BOTH, side=LEFT)
padding_right.pack(expand=True, fill=BOTH, side=RIGHT)
#padding_right2.pack(expand=True, fill=BOTH, side=RIGHT)

def installer_screen(window):
    #Destruim la finestra
    window.destroy()
    window = Tk()
    # Modifiquem el color de fons
    window.configure(background=BACKGROUND_COLOR)

    # Afegim els estils per a la barra de menú
    style = ttk.Style()
    style.theme_settings("default", theme)

    # Configurar la font del ttk Combobox, la part desplegable
    window.option_add('*TCombobox*Listbox.font',
tkFont.Font(family=TEXT_FONT, size=FONT_SIZE))

    # Afegim una icona
    icon = PhotoImage(file='odroid_cluster_icon.png')
    window.tk.call('wm', 'iconphoto', window._w, icon)

    # Maximitzem la finestra
    window.geometry("%dx%d+0+0" % (window.winfo_screenwidth(),
window.winfo_screenheight()))
    # Afegim el text de dalt
    window.winfo_toplevel().title("Odroid Cluster")
    add_menu(window)
    window.notebook.pack(expand=1, fill=BOTH)
    add_content_install(window)
    add_content_advanced(window)
    add_content_slurm(window)
def write_options(window):

    correct=True

    try:
        int(window.f2.upgrade_entry.get())
    except ValueError:
        messagebox.showerror(message='The upgrade time must be a digit.',
title="Upgrade time")
        correct=False

    if(int(window.f2.upgrade_entry.get()) < 0):
        messagebox.showerror(message='The upgrade time cannot be negative.',
title="Upgrade time")
        correct=False

    try:
        p=Path(window.f2.scripts_dir.get())
    except(SyntaxError, TypeError):
        messagebox.showerror(message='The scripts directory is invalid.',
title="Scripts dir")
        correct=False

    if(window.f2.scripts_dir.get()[0] != "/"):
        messagebox.showerror(message='Enter an absolute path for the scripts
directory.', title="Scripts dir")
        correct=False

    if(correct):
        correct = check_ip(window.f2.dns1, "DNS1")
    if(correct):
        correct = check_ip(window.f2.dns2, "DNS2")

    try:
        if(window.f3.max_time.get() != "INFINITE"):

```

```

        if(int(window.f3.max_time.get()) < 1):
            messagebox.showerror(message='Maximum time limit of jobs has
to be an integer value greater than 0 or INFINITE', title="Maximum time limit of jobs")
            correct=False
        except ValueError:
            messagebox.showerror(message='Maximum time limit of jobs has to be an
integer value greater than 0 or INFINITE', title="Maximum time limit of jobs")
            correct=False

    if(correct):
        write_option("DEFAULT_USER",window.f2.text_name.get())
        write_option("DEFAULT_PASSWORD",window.f2.text_password2.get())
        write_option("HOSTS_NAME",window.f2.text_hostname.get())
        write_option("EXTERNALDNS1",iplist_to_ipstring(window.f2.dns1))
        write_option("EXTERNALDNS2",iplist_to_ipstring(window.f2.dns2))
        write_option("SCRIPTS_DIR",window.f2.scripts_dir.get())
        write_option("UPGRADE",str(window.f2.check1.get()))
        write_option("IP_CLASS",str(window.f2.radiol.get()))
        write_option("SYS_TIMEZONE",str(window.f1.timezone.get()))

    write_option("SYS_LANGUAGE",str(icu[get_pos_list(window.f1.locale.get(),languages)])
)
        write_option("MASK", str(cidr_to_mask(mask.get())))
        write_option("IP", str(ip.ip_num1.get()) + "." + str(ip.ip_num2.get()) + "."
+ str(ip.ip_num3.get()) + "." + str(ip.ip_num4.get()))
        write_option("LAYOUT", layouts[layout.get()].get_code())
        write_option("UPGRADE_SLEEP", str(window.f2.upgrade_entry.get()))
        write_option("MAX_TIME", str(window.f3.max_time.get()))
        write_option("SCHEDULER",str(window.f3.scheduler.get()))
        write_option("TRACKING",str(window.f3.tracking.get()))
        if(window.f3.sched_type.get() == 0):
            write_option("SCHEDULER","sched/builtin")
        elif(window.f3.sched_type.get() == 1):
            write_option("SCHEDULER","sched/backfill")
        if(window.f3.track_type.get() == 0):
            write_option("TRACKING","proctrack/cgroup")
        elif(window.f3.track_type.get() == 1):
            write_option("TRACKING","proctrack/pgid")
        elif(window.f3.track_type.get() == 2):
            write_option("TRACKING","proctrack/linuxproc")
        for i in layouts[layout.get()].get_variants():
            if(variant.get() == i.get_name()):
                write_option("VARIANT", i.get_code())

    return correct

def start_installation(window):
    correct=write_options(window)
    if(correct):
        password=ask_password(window)
# Creem la finestra principal
root = Tk()
get_timezones()
get_languages()
get_layouts()
# Carrega la finestra de benvinguda
load_screen(root)
# Al cap de 1,5 segons carreguem el l'instalador
root.after(1500, lambda: installer_screen(root))
root.mainloop()

```