

Diseño y despliegue de un clúster de bajo presupuesto para el desarrollo de las prácticas de PSD

Daniel Quiñones Sánchez
Miguel Romero Martínez

Grado en Ingeniería de Computadores
Facultad de Informática



Universidad Complutense de Madrid

Curso Académico 2017/2018

Directores:

Alberto Núñez Covarrubias
Luis Llana Díaz

Agradecimientos

A mi compañero y amigo Rome, por aguantarme en todo momento y realizar la mayor parte del trabajo.

Daniel

A mi compañero y amigo Daniel, por aguantarme en todo momento y permitirme realizar la mayor parte del trabajo.

Miguel

Índice general

Índice	I
Índice de figuras	V
Resumen	VI
Abstract	VI
1.	2
2. Introducción	3
2.1. Introducción	3
2.2. Motivación	4
2.3. Trabajo relacionado	4
3. Arquitectura del Cluster	6
3.1. Componentes	6
3.2. Disposición de elementos	6
3.3. Montaje	6
3.4. Sistema centralizado	6
3.5. Problemas	6
3.5.1. Falta de energía	6
4. Configuración del clúster	8
4.1. Configuración del entorno de desarrollo	8

4.2. Montaje de servidores	9
4.3. Dynamic Host Configuration Protocol (DHCP)	10
4.3.1. Guía paso a paso de instalación de DHCP	11
4.4. Network File System (NFS)	13
4.4.1. Guía paso a paso de instalación de NFS	13
4.4.2. Creación y lanzamiento de servidor NFS como un daemon del sistema	15
4.5. Instalación de SIMCAN	15
4.5.1. Guía paso a paso de instalación en arquitectura ARM	16
4.6. Configuraciones derivadas de la arquitectura	17
4.6.1. Modificación del GRUB	17
4.7. Seguridad	18
4.8. Eliminación de usuarios y permisos	18
5. Diseño de prototipos	20
5.1. Diseño del Modelo 1	21
5.1.1. Modelado 3D	21
5.1.2. Resultados	22
5.2. Diseño del Modelo 2	24
5.2.1. Modelado 3D	24
5.2.2. Resultados	25
5.3. Diseño del Modelo 2b	27
5.3.1. Modelado 3D	27
5.4. Diseño del prototipo C	29
5.4.1. Modelado 3D	29
5.4.2. Resultados	29

6. Resultados Test	33
6.1. Introducción	33
6.2. Prueba 1	33
6.2.1. Escenario	33
6.2.2. Resultados	34
6.2.3. Conclusiones	34
6.3. Prueba 2	35
6.3.1. Escenario	35
6.3.2. Resultados	35
6.3.3. Conclusiones	36
6.4. Prueba 3	36
6.4.1. Escenario	36
6.4.2. Resultados	37
6.4.3. Conclusiones	37
6.5. Conclusiones generales	38
Bibliografia	39

Índice de figuras

4.1. Sistema de archivos	9
4.2. Esquema	10
4.3. Servidor DHCP	11
5.1. Modelo 1.1 3D	21
5.2. Modelo 1.2 3D	21
5.3. Modelo 1.3 3D	22
5.4. Resultado Modelo 1.1	22
5.5. Resultado Modelo 1.2	23
5.6. Resultado Modelo 1.3	23
5.7. Modelo 2.1 3D	24
5.8. Modelo 2.2 3D	24
5.9. Modelo 2.3 3D	25
5.10. Resultado Modelo 2.1	25
5.11. Resultado Modelo 2.2	26
5.12. Resultado Modelo 2.3	26
5.13. Resultado Modelo 2.4	27
5.14. Modelo 2b.1 3D	27
5.15. Modelo 2b.2 3D	28
5.16. Modelo 2b.3 3D	28
5.17. Resultado Modelo 3.1	29
5.18. Resultado Modelo 3.2	30

5.19. Resultado Modelo 3.3	31
5.20. Resultado Modelo 3.4	31
6.1. Prueba 1, Modelo 1	34
6.2. Prueba 1, Modelo 2	34
6.3. Prueba 1, Modelo 2b	34
6.4. Prueba 2, Modelo 1	35
6.5. Prueba 2, Modelo 2	35
6.6. Prueba 2, Modelo 2b	36
6.7. Prueba 3, Modelo 1	37
6.8. Prueba 3, Modelo 2	37
6.9. Prueba 3, Modelo 2b	37

Capítulo 1

*Tendrá todo el dinero del mundo, pero hay algo
que nunca podrá comprar. . . un dinosaurio.*

Homer J. Simpson

*No te engañes Jimmy. Si una vaca tuviera la
oportunidad, te comería a ti, y a los seres que tú
mas quieres.*

Troy McClure

Capítulo 2

Introducción

*La cosa está muy mal... estoy friendo los huevos
con saliva*

Gregorio Esteban Sánchez

2.1. Introducción

El objetivo principal de este trabajo es el desarrollo de un cluster de cómputo de bajo presupuesto. Para ello hemos echo uso de los denominados computadores de placa reducida, en concreto Raspberry Pi, como nodos de procesamiento sobre una distribución de linux específica para estas. Nos planteamos los siguientes objetivos específicos:

- Diseño de la caja
- Distribución de cada uno de los elementos dentro de esta
- Estudio de las temperaturas y comportamiento del Hardware bajo situaciones de stress
- Despliegue y desarrollo del software del sistema
- Generación de guías de instalación y aspectos específicos de configuración del sistema
- Desarrollo de un planificador de tareas para la distribución de trabajos entre los nodos
- Desarrollo de un software para realizar el envío y recepción de trabajos al cluster

2.2. Motivación

En los últimos años se ha incrementado ese tipo de proyectos por parte de distintas universidades y proyectos de particulares. Principalmente estos se han centrado únicamente en aumentar el número de dispositivos, pero no hay un estudio de comportamiento real de uno de ellos, y tampoco una guía con la que poder establecer los pasos a seguir, nuestro propósito ha sido el de monitorizar todos los aspectos del desarrollo de uno de estos clusters desde el punto de vista de la correcta disposición y distribución de los componentes para mejorar aspectos como la accesibilidad, la correcta refrigeración de los nodos, así como la mejora del rendimiento en ellos para finalmente obtener un cluster de cómputo de bajo coste que pueda ser viable.

2.3. Trabajo relacionado

Existen multitud de proyectos de este tipo, destacaremos los mas interesantes:

- [VMW Research Group Raspberry Pi Cluster](#), dispone de un cluster de 24 nodos realizado con raspberry Pi 2 con una interfaz de pantalla táctil y dos adaptadores de Ethernet que controlan la fuente de alimentación , sirve DHCP, NFS.
- [Afkham Azeez](#), ha creado un cluster de 16 nodos distribuidas en un módulo de aluminio de dos niveles.
- [Universidad de Southampton](#), investigadores de esta universidad han construido una supercomputadora de Raspberry Pi unidas con Lego. El profesor Simon Cox y su equipo construyeron la supercomputadora de 64 procesadores y 1 TB de memoria. Tiene un coste aproximado de 3100€.
- [David Guill](#), en su web Like Magic Appears ofrece una guía de construcción de un cluster de 40 nodos y dispone de material audiovisual como guía.

Capítulo 3

Arquitectura del Cluster

*Tenemos que fabricar máquinas que nos
permitan seguir fabricando máquinas, porque lo
que no va a hacer nunca la máquina es fabricar
máquinas*

M.Rajoy

3.1. Componentes

3.2. Disposición de elementos

3.3. Montaje

3.4. Sistema centralizado

3.5. Problemas

3.5.1. Falta de energía

Capítulo 4

Configuración del clúster

'long long long' is too long for GCC

Some GCC programmer

4.1. Configuración del entorno de desarrollo

Debido al volumen del material necesario para el desarrollo del proyecto, la necesidad de realizar el montaje y desmontaje de forma manual de éste y la dificultad en el acceso y configuración de cada uno de los nodos que lo componen decidimos invertir tiempo en realizar una virtualización del sistema para minimizar los problemas antes descritos. Así, al disponer de un entorno virtual, que replica el cluster real, se minimiza el tiempo necesario para la configuración de los distintos servicios y servidores del sistema operativo y sirve como banco de pruebas para el desarrollo software.

Para ello hemos utilizamos VMware workstation 12 como plataforma de software de virtualización y una ISO de Raspbian basada en Debian Stretch disponible en la página oficial de Raspberry Pi. Aunque el sistema del cluster parte de una versión diferente de debian el sistema de carpetas y sobre todo la instalación de SIMCAN es similar al del entorno real.

En el repositorio en github existe una réplica del sistema de carpetas tanto para el servidor como para los nodos esclavo. Cada una de las carpetas y ficheros modificados en el

sistema durante la configuración del sistema quedan reflejados en el repositorio, disponiendo así de un listado en forma de árbol de todas las configuraciones necesarias para el correcto funcionamiento de este.

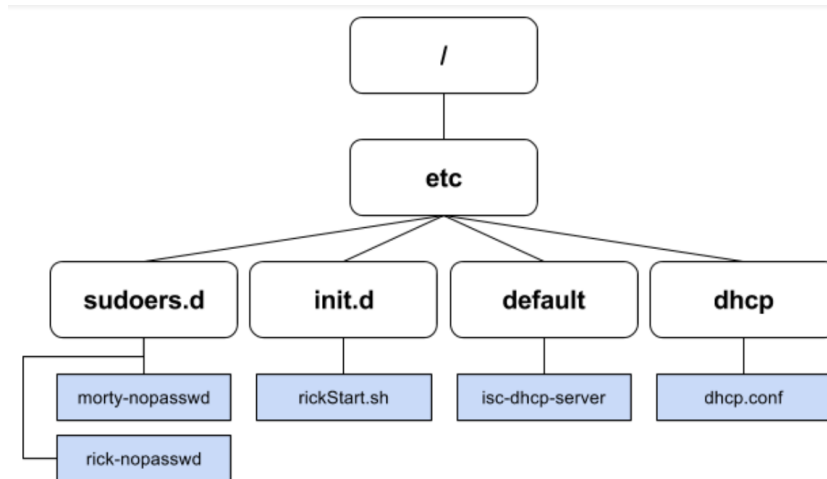


Figura 4.1: Sistema de archivos del repositorio

4.2. Montaje de servidores

El cluster está dividido en dos partes bien diferenciadas, por un lado, el front-end, donde se dispone de un único servidor, el cual es el único nodo del cluster que tiene instalado el software de SIMCAN además de ofrecer los servicios de NFS, DHCP y SSH entre otros como muestra la figura XX. Éste además es el punto de comunicación con el exterior, al disponer de una tarjeta de red adicional.

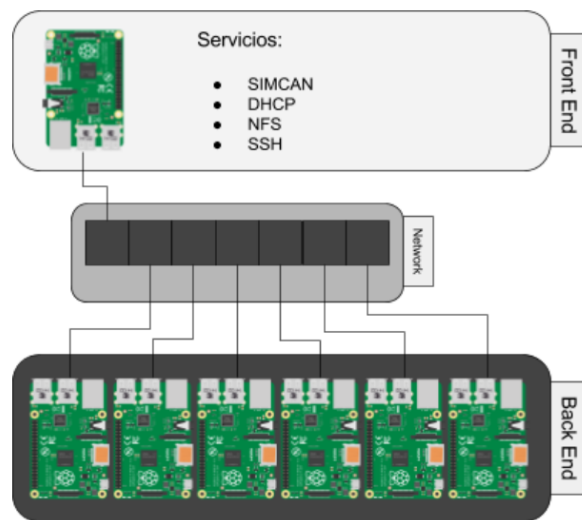


Figura 4.2: Esquema del sistema

Por otro lado, en el back-end, disponemos de varios nodos esclavo que disponen únicamente del sistema Debian Jessie y unas pocas librerías necesarias para realizar las operaciones de computo.

4.3. Dynamic Host Configuration Protocol (DHCP)

El nodo maestro hará de servidor repartiendo las direcciones de forma dinámica al resto de clientes. La red para del back-end es la 172.16.111.0/24, el nodo maestro tiene asignada de forma manual la dirección 172.16.111.1/24 y hace un reparto con el resto de direcciones del rango como se muestra en la figura XX. El montaje de DHCP evita tener que asignar manualmente direcciones a todos los nodos, sin embargo es más útil en la práctica que cada uno de los nodos disponga de la misma dirección el máximo tiempo posible, para ello se han modificado el parámetro `default-lease-time`, que determina el tiempo de concesión de una IP cuando el cliente en su solicitud no lo especifica, del fichero `/etc/dhcp/dhcp.conf` a sus máximos valores posibles. Estableciendo un valor de 7776000 segundos en dicho campo

obtenemos un tiempo de concesión de noventa días.

Los servicios ofrecidos por el nodo maestro de cara al front-end están disponibles a través de su segunda tarjeta de red, configurada para obtener una dirección IP desde un ISP.

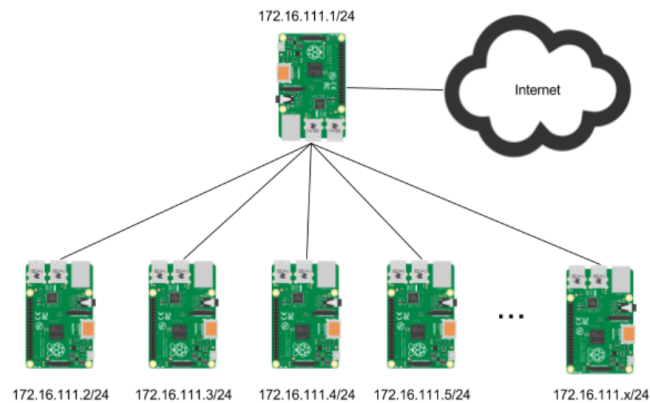


Figura 4.3: Servidor DHCP

4.3.1. Guía paso a paso de instalación de DHCP

Instalación de paquetes en el servidor

```
1 sudo apt-get update
2 sudo apt-get install isc-dhcp-server
```

Editar fichero `/etc/default/isc-dhcp-server`

```
3 sudo nano /etc/default/isc-dhcp-server
4 INTERFACES = eth0
```

Editar fichero `/etc/network/interfaces`

```
5 nano /etc/network/interfaces
6 source-directory /etc/network/interfaces.d

auto eth0
iface eth0 inet static
address 172.16.111.106
```

```
netmask 255.255.255.0
network 192.168.100.0
gateway 172.16.111.105

allow-hotplug wlan0
iface wlan0 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

allow-hotplug wlan1
iface wlan1 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

Editar fichero de configuración /etc/dhcp/dhcpd.conf

Incluir la siguiente configuración al final del fichero:

```
subnet 172.16.111.0 netmask 255.255.255.0{
    range 172.16.111.106 172.16.111.255;
    option domain-name-servers 8.8.8.8, 4.4.4.4;
    default-lease-time 7776000;
    option routers 172.16.111.105;
}

host Rick{
    hardware ethernet <dir MAC del servidor>
    fixed-address 172.16.111.110;
}
```

Activación de la red eth0 y reinicio

```
7 sudo ifconfig eth0 up
8 sudo reboot now
```

Arranque del servicio dhcp

```
9 sudo /usr/sbin/dhcpd
```

Comprobación del correcto funcionamiento del sistema

```
10 ps -ef | grep dhcpd
```

Opcionalmente se pueden mostrar las máquinas conectadas al servicio mediante la orden

```
11 cat /var/lib/dhcp/dhcp.leases
```

4.4. Network File System (NFS)

Como se destacaba anteriormente, el nodo servidor es el único que dispone de una versión de SIMCAN instalada, esta configuración ofrece la posibilidad de que la labor de los nodos esclavo sea únicamente la de realizar el procesamiento de datos. Mediante NFS, el nodo servidor comparte su carpeta home durante el arranque del sistema, de esta forma, el resto de nodos esclavo realizan el montaje de este home compartido en red en su propio directorio home, creando así un único punto de acceso compartido en red del que se pueden extraer los ejecutables sin la necesidad de disponer de SIMCAN instalado. El maestro se encarga de realizar la compilación de los ficheros .ned y pone a disposición del resto de nodos los ejecutables.

Es necesario que todos los nodos de la red tengan un mismo usuario común para conseguir una correcta sincronización, de igual manera hay que mantener un estricto control de los permisos de cada uno de los nodos esclavo tanto a nivel interno como de cara al servidor.

4.4.1. Guía paso a paso de instalación de NFS

Instalar paquetes en el servidor

```
1 sudo apt-get update
2 sudo apt-get install nfs-kernel-server
```

Editar fichero de configuración /etc/exports en servidor

```
Incluir al final del fichero el directorio a compartir:
Ruta de carpeta
Dirección IP de máquina destino / permisos
```

```
Ejemplo: /home/morty 172.16.111.0/24(rw,no_subtree_check)
```

Instalación de paquetes en el cliente

```
Instalados por defecto en raspBian
3 sudo apt-get update
4 sudo apt-get install build-essential gcc g++ bison flex perl tcl-dev tk-
  dev libxml2-dev zlib1g-dev default-jre doxygen graphviz libwebkitgtk-1.0-0
  openmpi-bin libopenmpi-dev libpcap-dev
```

Ejecutar cambios realizados en el servidor

```
5 sudo exportfs -a
6 sudo mount 172.16.111.x:/home/morty /home/morty
  172.16.111.x es la dirección ip de servidor
```

Reiniciar servicios

```
7 sudo /etc/init.d/rpcbind restart
8 sudo /etc/init.d/nfs-kernel-server restart
```

PROBLEMAS:

Permisos desde la maquina cliente:

```
El usuario de la máquina cliente ha de ser sudoer
La carpeta compartida debe tener permisos de lectura , escritura y ejecución
9 chmod -R 0777 escenario
```

A fin de evitar tener que realizar el arranque del servidor de forma manual es recomendable crear un daemon he incluirlo en el directorio /etc/init.d para que se ejecute al arranque del sistema de forma automática.

4.4.2. Creación y lanzamiento de servidor NFS como un daemon del sistema

Poner algo de texto explicando bla bla bla

Contenido del script:

```
#!/bin/bash
### BEGIN INIT INFO
# Provides: M.Romero && D.Quinones
# Required-start: $syslog
# Required-stop: $syslog
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: Inicialización de servicios nfs
# Description:
### END INIT INFO

sudo exportfs -a
sudo /etc/init.d/rpcbind restart
sudo /etc/init.d/nfs-kernel-server restart

# El orden de estos últimos comandos es esencial
```

Creación y lanzamiento del un daemon

```
10 chmod +x rickStart.sh
11 cp rickStart.sh /etc/init.d/
12 cd /etc/init.d
13 update-rc.d rickStart.sh defaults
```

4.5. Instalación de SIMCAN

Antes de poder instalar el software SIMCAN es necesario realizar la instalación previa del simulador modular de eventos discretos de redes Omnet++ en su versión 4.6. Además de la suite Inet, que implementa modelos de código abierto OMNeT++ para redes cableadas, inalámbricas y móviles.

Debido a la baja potencia de la Raspberry esta no es capaz de lanzar la aplicación de

forma gráfica, esto supone un problema a la hora de realizar la instalación del software. Es por esto que todas las instalaciones han de realizarse de forma manual a través del terminal. Esto afecta principalmente a la instalación de Inet, ya que las principales guías de instalación disponibles en las webs oficiales parten siempre del entorno gráfico de Omnet++.

Durante el desarrollo del proyecto hemos generado unas guías de instalación y configuración paso a paso que se desglosarán en el siguiente apartado.

4.5.1. Guía paso a paso de instalación en arquitectura ARM

Descargar los tar.gz de Omnet 4.6, Inet, simcan.tar. Esta última (simcan) incluye las bibliotecas que se necesitan para la compilación. Copia los archivos .tar de Omnet e Inet en /pi y se descomprimen Desde el directorio /pi ejecuta los siguientes comandos

Instalación de Omnet

```
1 sudo apt-get update
2 sudo apt-get install build-essential gcc g++ bison flex perl tcl-dev tk-
  dev libxml2-dev zlib1g-dev default-jre doxygen graphviz libwebkitgtk-1.0-0
  openmpi-bin libopenmpi-dev libpcap-dev
3 sudo apt-get install gnome-color-chooser
4 cd omnetpp-4.6
5 . setenv
6 ./configure
7 make
```

Instalación de Inet

Crear un directorio nuevo en /omnet-4.6 llamado project
Copiar en el directorio inet descomprimido y ejecutar:

```
8 sudo apt-get install libavcodec-dev libavformat-dev
9 make makefiles
10 make
```

Instalación de Simcan

Copia el directorio de simcan a projects y ejecuta los siguientes comandos:


```
11 export omnetpp_root=$HOME/morty/omnnetpp-4.6
12 export INET_HOME=$omnetpp_root/projects/inet
13 export SIMCAN_HOME=$omnetpp_root/projects/simcan
14 export LD_LIBRARY_PATH=$omnetpp_root/lib:$LD_LIBRARY_PATH
15 export PATH=omnetpp_root/bin:$PATH
16 make makefiles
17 make
```

Enjoy!

4.6. Configuraciones derivadas de la arquitectura

Como se ha explicado anteriormente la arquitectura elegida distribuye la potencia a todos los nodos del cluster desde una misma fuente de alimentación. Debido a la menor cantidad de recursos y servicios que han de ofrecer los nodos esclavo, éstos tienen una carga del sistema ligeramente más rápida que la del nodo maestro, debido a ello, en alguno se pueden producir problemas de sincronización de servicios, más concretamente con el montaje de sistemas de ficheros compartidos en red por NFS, el cual es crítico para el funcionamiento general del sistema. Para solucionar este problema se han realizado modificaciones a fin de conseguir acelerar la carga del nodo maestro y ralentizar la del resto.

4.6.1. Modificación del GRUB

Una de las soluciones más sencillas para resolver el problema de sincronización consiste en aumentar el tiempo por defecto del grub de los nodos esclavo ya que la carga del sistema se produce cuando éste termina. Para ello únicamente hay que modificar la opción GRUBTIMEOUT en el fichero `/etc/default/grub`. En las pruebas realizadas durante la virtualización del sistema se comprobó que estableciendo un retardo de veinticinco segundos era suficiente para que el nodo maestro realizase la carga completa del sistema. Sin embargo queda por comprobar que en el entorno real esto se sigue produciendo.

Contenido del fichero /etc/default/grub de un nodo esclavo:

```
GRUB_DEFAULT=0
GRUB_TIMEOUT=25
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash plymouth.ignore-serial- consoles"
GRUB_CMDLINE_LINUX=""
```

4.7. Seguridad

Por defecto, en las distribuciones de Debian Jessie existentes en los repositorios oficiales de raspberry.org existe... Vienen configurados usuarios por defecto Root no tiene contraseña Hay que eliminar a pi como superuser Sólo dejar morty como superuser en maestro pero no en esclavos

4.8. Eliminación de usuarios y permisos

UID, todos han de tener el mismo, permisos en maestro, esclavo Inicialización del sistema mediante scripts

Capítulo 5

Diseño de prototipos

*Bueno, pero aparte del alcantarillado, la sanidad,
la enseñanza, el vino, el orden público, la
irrigación, las carreteras y los baños públicos,
¿qué han hecho los romanos por nosotros?*

The Life Of Brian

5.1. Diseño del Modelo 1

5.1.1. Modelado 3D

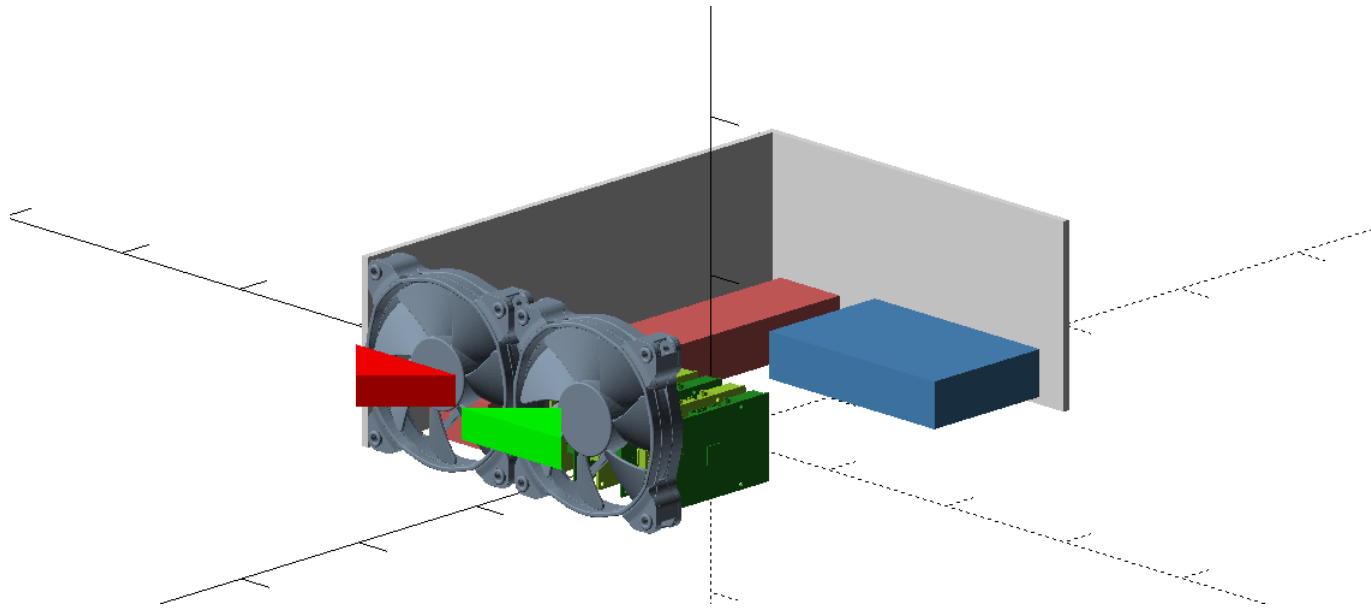


Figura 5.1: Modelo 1.1 3D

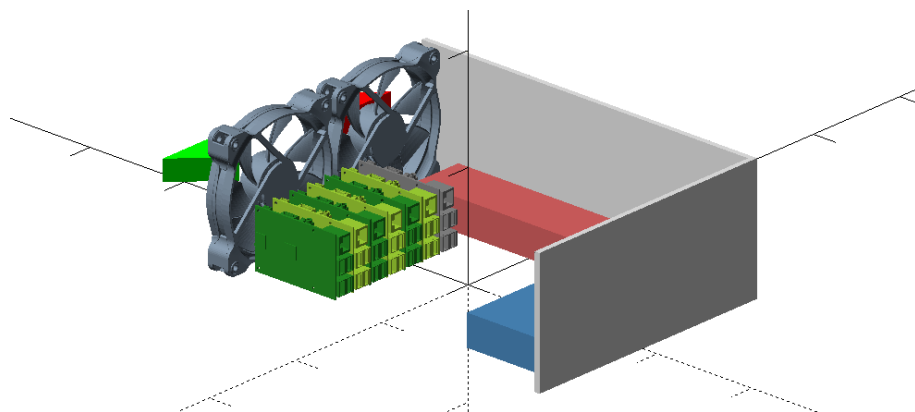


Figura 5.2: Modelo 1.2 3D

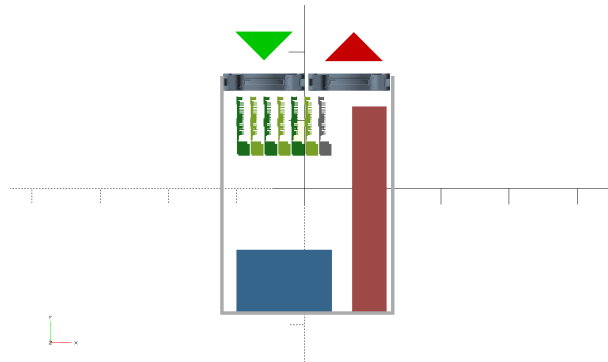


Figura 5.3: Modelo 1.3 3D

5.1.2. Resultados



Figura 5.4: Resultado M1.1



Figura 5.5: Resultado M1.2



Figura 5.6: Resultado M1.3

5.2. Diseño del Modelo 2

5.2.1. Modelado 3D

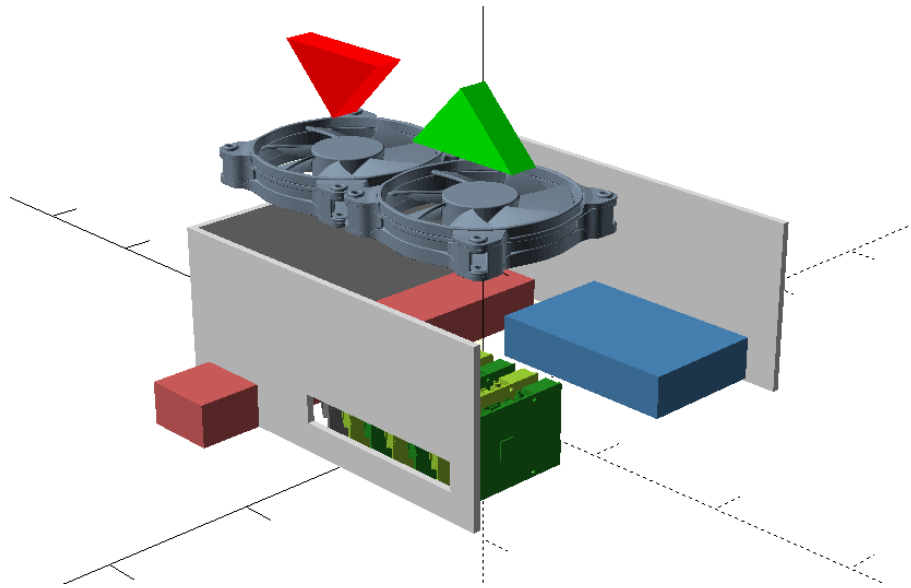


Figura 5.7: Modelo 2.1 3D

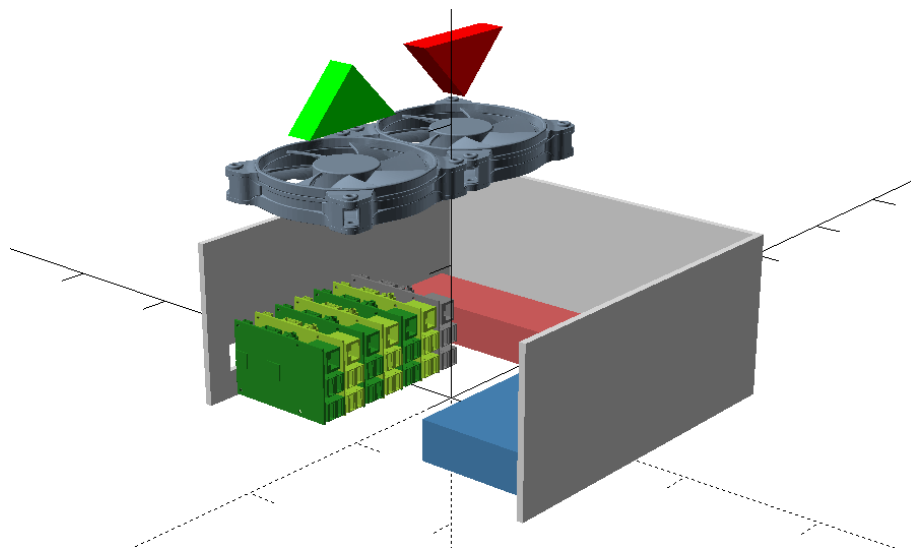


Figura 5.8: Modelo 2.2 3D

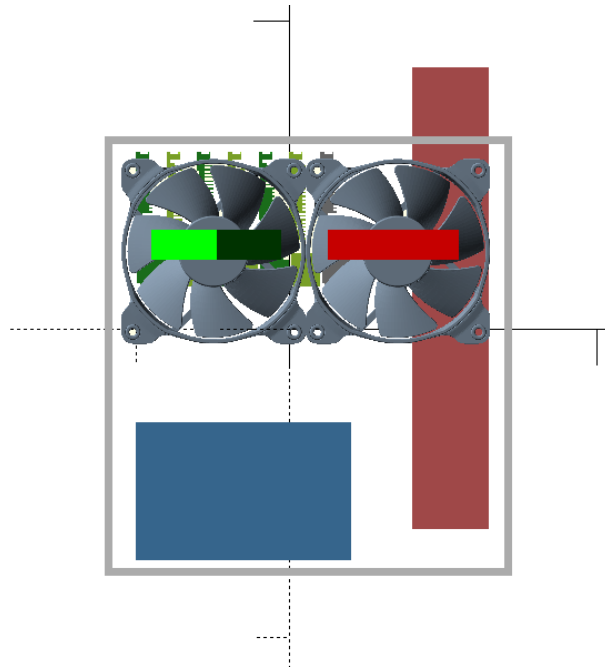


Figura 5.9: Modelo 2.3 3D

5.2.2. Resultados

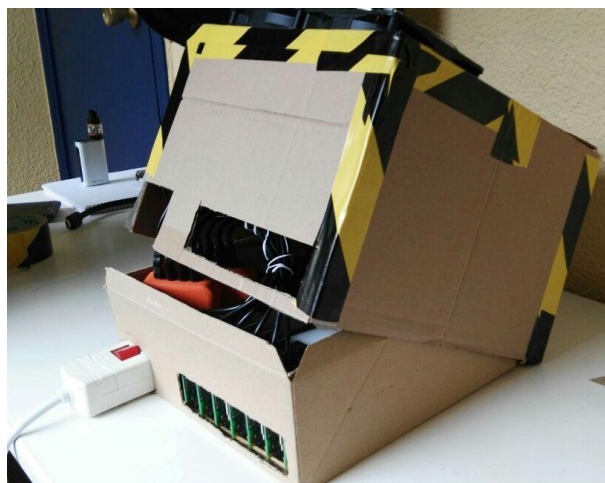


Figura 5.10: Resultado M2.1



Figura 5.11: Resultado M2.2



Figura 5.12: Resultado M2.3



Figura 5.13: Resultado M2.4

5.3. Diseño del Modelo 2b

5.3.1. Modelado 3D

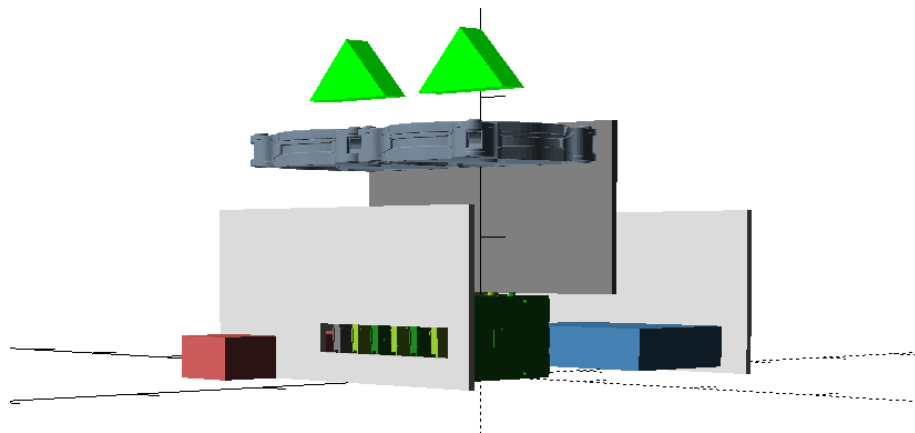


Figura 5.14: Modelo 2b.1 3D

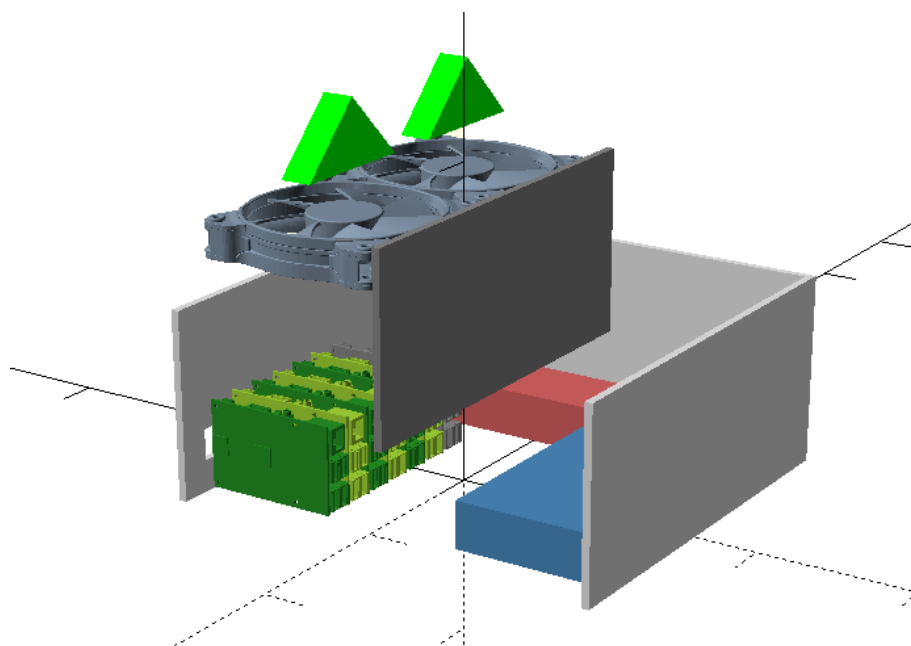


Figura 5.15: Miau

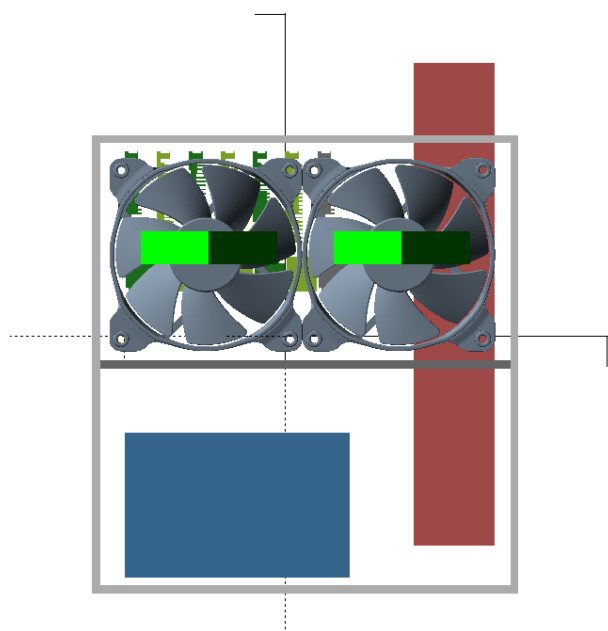


Figura 5.16: Modelo 2b.3 3D

5.4. Diseño del prototipo C

5.4.1. Modelado 3D

5.4.2. Resultados



Figura 5.17: Resultado M3.1



Figura 5.18: Resultado M3.2



Figura 5.19: Resultado M3.3

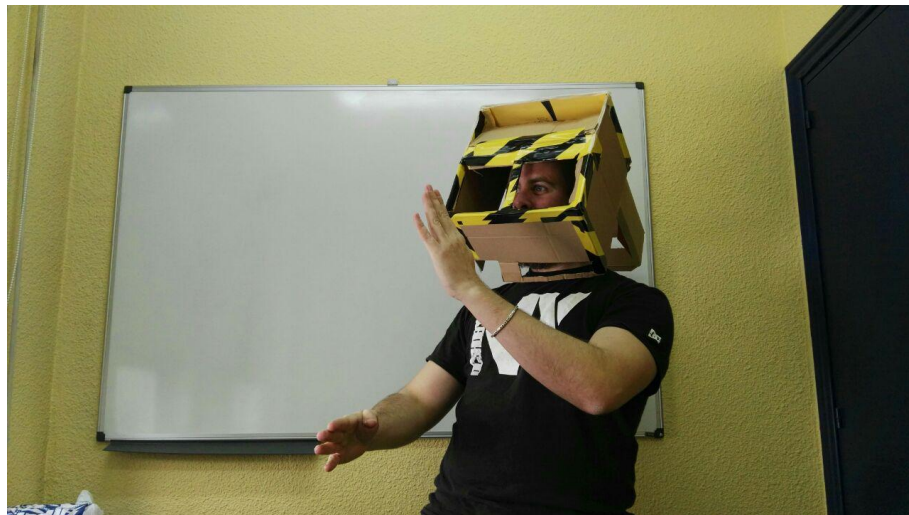


Figura 5.20: Destruir.a.los.HUMANOS

Capítulo 6

Resultados Test

*I too have a Nuclear Button, but it is a much
bigger and more powerful one than his, and my
Button works!*

Donald J. Trump

6.1. Introducción

6.2. Prueba 1

6.2.1. Escenario

- Toma: 10 segundos
- Número de raspberries: 1
- Número de Cores: 4
- Tiempo: 3 horas
- Temperatura ambiente comienzo: 14/16
- Temperatura ambiente fin: 16/18

6.2.2. Resultados

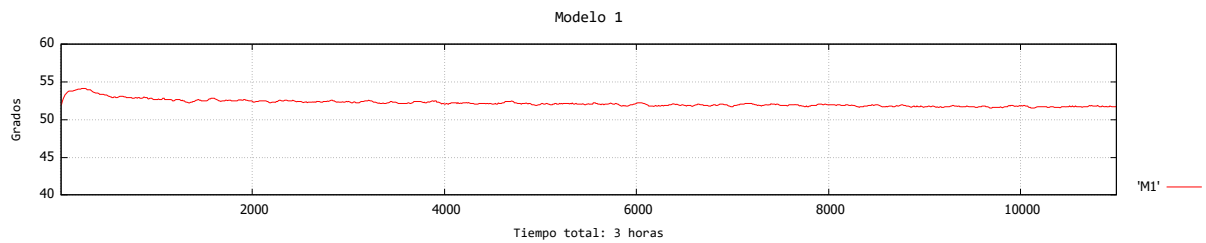


Figura 6.1: Modelo 1

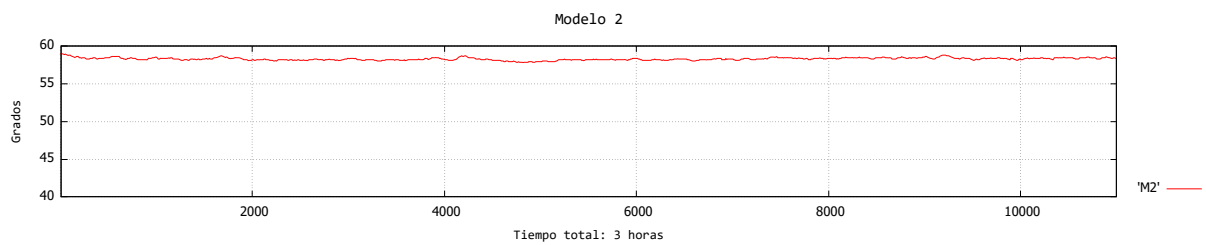


Figura 6.2: Modelo 2

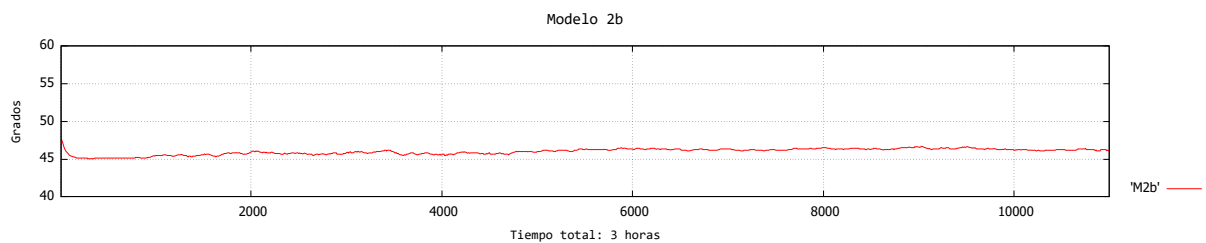


Figura 6.3: Modelo 2b

6.2.3. Conclusiones

Se puede comprobar que el modelo 2b es el que mantiene una mejor tasa de temperatura respecto a los demás, en todo caso el sistema se mantiene estable durante todo el periodo, con muy poca variación en su temperatura máxima y mínima.

El modelo 2, aunque es muy estable, registra unos valores cercanos a los 60° , con lo que es sin duda el peor de los tres.

6.3. Prueba 2

6.3.1. Escenario

- Toma: 10 segundos
- Número de raspberries: 3
- Número de Cores: 4
- Tiempo: 3 horas
- Temperatura ambiente comienzo: 15/16
- Temperatura ambiente fin: 16/18

6.3.2. Resultados

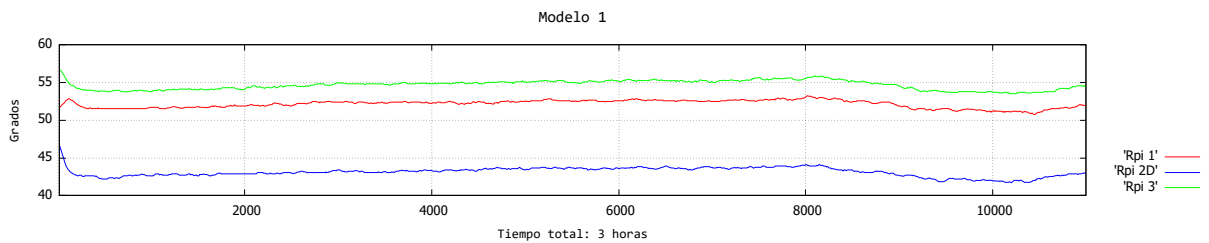


Figura 6.4: Modelo 1

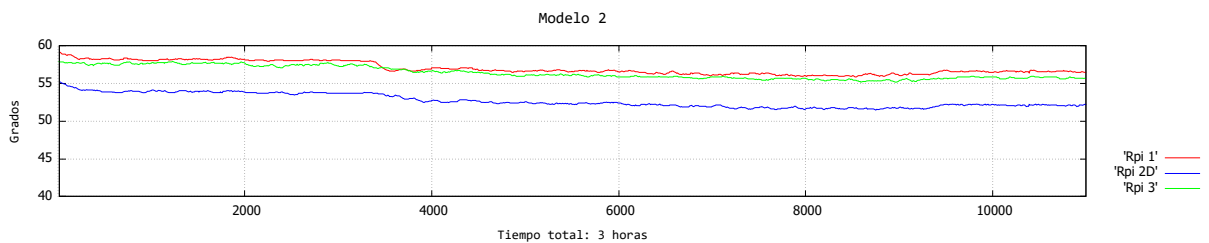


Figura 6.5: Modelo 2

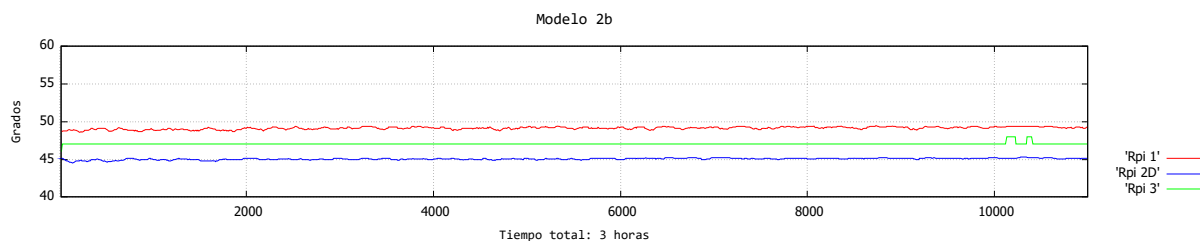


Figura 6.6: Modelo 2b

6.3.3. Conclusiones

En esta prueba se incluye un nodo el cual dispone de un disipador de calor, hemos comprobado que una buena corriente de aire hace que aquellos nodos que disponen de esta mejora presenten unos resultados notablemente inferiores al resto. Sin embargo, como pasa en el modelo 2 , si la corriente no es eficiente dicha mejora se vé sevéramente afectada, aunque sigue siendo notable.

Nuevamente el modelo 2b ofrece unos resultados mejores al resto, además de ofrecer ser mucho mas estables durante toda la prueba. Al contrario que con el modelo 2 en este, debido a la eficiencia en el encaminamiento del flujo de aire en el contenedor.

6.4. Prueba 3

6.4.1. Escenario

- Toma: 10 segundos
- Número de raspberries: 6
- Número de Cores: 4
- Tiempo: 3 horas
- Temperatura ambiente comienzo: 16/17
- Temperatura ambiente fin: 14/15

6.4.2. Resultados

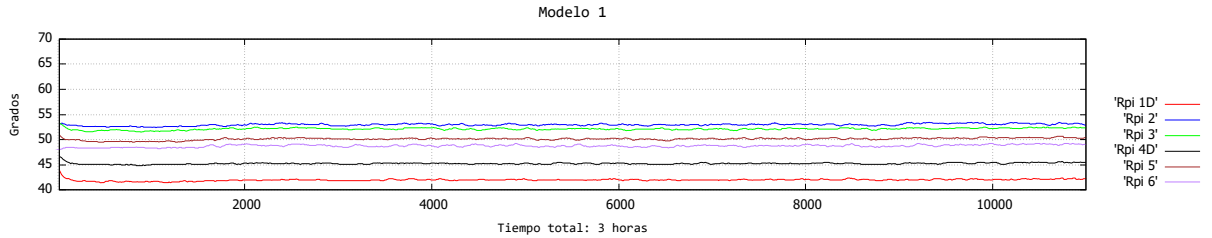


Figura 6.7: Modelo 1

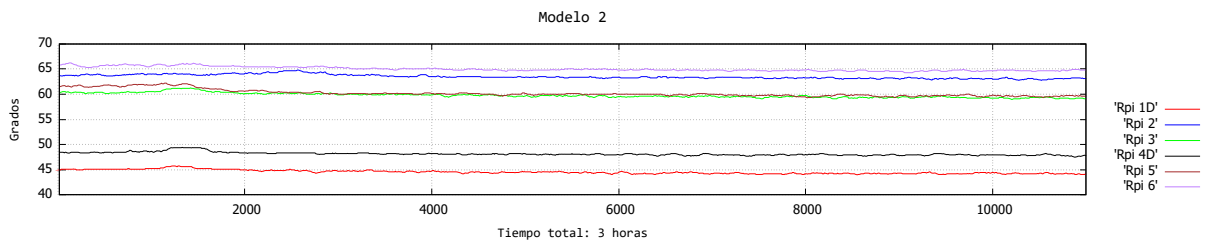


Figura 6.8: Modelo 2

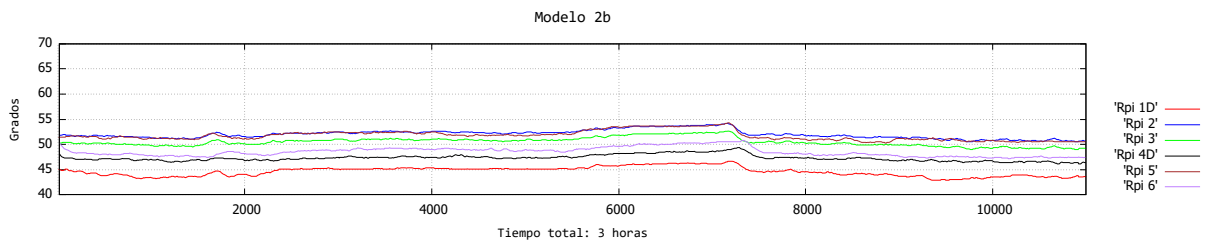


Figura 6.9: Modelo 2b

6.4.3. Conclusiones

En esta prueba hemos detectado que definitivamente el modelo 2 no cumple con las expectativas necesarias para el correcto funcionamiento del cluster, la diferencia entre nodos con y sin disipador son muy notables y aquellas que no disponen de esta mejora arrojan unas temperaturas muy próximas a la temperatura crítica de parada, por lo que podemos descartar este modelo para el desarrollo final.

Los otros dos modelos ofrecen unos resultados muy semejantes a los de las anteriores pruebas, particularmente el modelo 1 sigue ofreciendo mayor mejora entre los nodos con disipador y el modelo 2b, aunque ofrece algunas variaciones mas notables durante la prueba sigue manteniendo una relación parecida entre nodos, independientemente de que si incluyan o no mejoras de disipación de calor.

6.5. Conclusiones generales

Durante el desarrollo de los test hemos podido aprender cuales son los diferentes factores que influyen sobre el sistema. Las corrientes de aire tienen una gran afección, con lo que merece la pena dedicar gran parte del trabajo en buscar una eficiencia en la conducción de flujos de aire dentro del receptáculo, eso nos permite reducir el número de ventiladores y establecer un punto de partida para la disposición del resto de elementos.

Al contar con un gran número de cables, también hemos tratado de combinar el punto anterior con la accesibilidad general a los dispositivos, de esta manera, exponiendo las tarjetas SD conseguimos que no sea necesario abrir el receptáculo para acceder a cada uno de los nodos.

La temperatura ambiente es otro de los factores influyentes en el comportamiento general, aunque menos notorio, se pueden apreciar alteraciones en todos los dispositivos sujetos a este efecto. sería preciso que la temperatura ambiente se mantuviera lo mas estable posible.

En cuanto a la robustez, se han llevado a cabo pruebas de estrés de 24 horas, en todas ellas el comportamiento ha sido similar a las de menor número de horas, por lo que podemos concluir que el sistema es estable y poco propenso a las caídas repentinas por periodos largos de trabajo.

En las diferentes pruebas realizadas hemos observado que no existe una correlación entre tasa de trabajo completado y temperatura. Aún así, con el fin de prolongar la vida útil de cada nodo, siempre es mejor mantener un rango de temperaturas bajo.

El uso de disipadores es muy conveniente, el precio de estos es muy reducido, y con una distribución adecuada aumentan tanto la vida útil del cluster como el comportamiento de los nodos ante los cambios bruscos de temperatura ambiente, en el peor de los casos, todos los nodos que disponían de esta mejora ofrecieron unos resultados mejor a la del resto.