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



TRABAJO FIN DE GRADO
Ingeniería de Computadores

Dirigido por

Alberto Núñez Covarrubias

Codirigido por

Luis Llana Díaz

Daniel Quiñones Sánchez
Miguel Romero Martínez

Departamento de Sistemas Informáticos y Computación
Facultad de Informática
Universidad Complutense de Madrid

Junio 2018

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

Memoria de Trabajo de Fin de Grado

**Daniel Quiñones Sánchez
Miguel Romero Martínez**

Dirigido por

Alberto Núñez Covarrubias

**Departamento de Sistemas Informáticos y Computación
Facultad de Informática
Universidad Complutense de Madrid**

Junio 2018

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

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

Agradecimientos

Resumen

Palabras clave

Siglas

Abstract

Keywords

Índice

Agradecimientos	IX
Resumen	X
Palabras clave	XI
Siglas	XII
Abstract	XIII
Keywords	XIV
1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	1
1.3. Estructura	1
1.4. Motivation	1
1.5. Goals	1
	XV

1.6. Structure	1
2. Trabajo relacionado	2
2.1. Nuestra aportación	2
3. Arquitectura del Cluster	3
3.1. Componentes	4
3.2. Diseño del prototipo A	4
3.2.1. Modelado 3D	4
3.2.2. Resultados (fotos)	4
3.3. Diseño del prototipo B	4
3.3.1. Modelado 3D	4
3.3.2. Resultados (fotos)	4
3.4. Diseño del prototipo C	4
3.4.1. Modelado 3D	4
3.4.2. Resultados (fotos)	4
3.5. Evaluación de prototipos A,B,C	4
3.5.1. Rendimiento prototipos A,B,C	4
3.5.2. Temperatura prototipos A,B,C	4
3.6. Problemas	4
3.6.1. Falta de energía	4
4. Configuración del cluster	5

4.1. Virtualización del Sistema	5
4.1.1. Raspbian y sus vicisitudes	6
4.2. Montaje de servidores	6
4.2.1. DHCP	6
4.2.2. NFS	6
4.2.3. SSH	6
4.3. Instalación de SIMCAN	6
4.3.1. Guía paso a paso de instalación en arquitectura ARM	7
4.4. Optimización y rendimiento	7
4.4.1. Modificación del GRUB	7
4.4.2. Paralelización del arranque en servidor	7
4.5. Seguridad	7
4.5.1. Eliminación de usuarios y permisos	7
4.6. Inicialización del sistema mediante scripts	7
5. Desarrollo del software	8
6. Pruebas	9
6.1. Comparación con otros procesadores de mayor potencia	9
6.2. Estudio a partir de qué nivel de uso del cluster merece la pena ante un procesador normal	9
6.3. Pruebas contra el cluster de Amazon(???)	9

7. Trabajo individual	10
7.1. Miguel Romero Martínez:	10
7.2. Daniel Quiñones Sánchez:	10
 8. Conclusiones y trabajo futuro	 11
8.1. Conclusiones generales	11
8.2. Trabajo futuro	11
8.3. General conclusions:	11
8.4. Future work:	11
 Bibliografía	 13

Índice de figuras

4.1. Estructura en árbol	6
5.1. Raspberry, te das cuen	8

Índice de Tablas

Capítulo 1

Introducción

"long long long is too long for GCC"
some programmer from GCC project

1.1. Motivación

1.2. Objetivos

1.3. Estructura

1.4. Motivation

1.5. Goals

1.6. Structure

Capítulo 2

Trabajo relacionado

*Me gusta y me fascina el trabajo. Podría estar sentado
horas y horas mirando a otros cómo trabajan.*

Jerome K. Jerome, Humorista inglés.

2.1. Nuestra aportación

Capítulo 3

Arquitectura del Cluster

3.1. Componentes

3.2. Diseño del prototipo A

3.2.1. Modelado 3D

3.2.2. Resultados (fotos)

3.3. Diseño del prototipo B

3.3.1. Modelado 3D

3.3.2. Resultados (fotos)

3.4. Diseño del prototipo C

3.4.1. Modelado 3D

3.4.2. Resultados (fotos)

3.5. Evaluación de prototipos A,B,C

Capítulo 4

Configuración del cluster

4.1. Virtualización del Sistema

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 slave. 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.

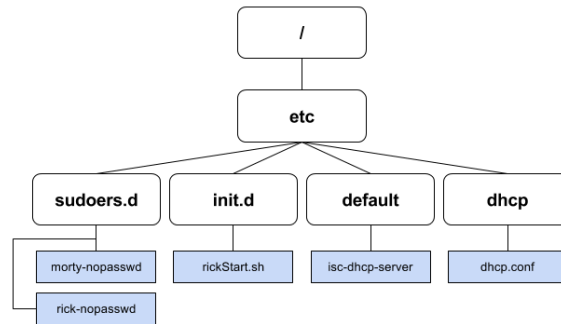


Figura 4.1: Estructura en árbol

4.1.1. Raspbian y sus vicisitudes

4.2. Montaje de servidores

4.2.1. DHCP

4.2.2. NFS

4.2.3. SSH

4.3. 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.

4.3.1. 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

4.4. Optimización y rendimiento

4.4.1. Modificación del GRUB

4.4.2. Paralelización del arranque en servidor

4.5. Seguridad

4.5.1. Eliminación de usuarios y permisos

4.6. Inicialización del sistema mediante scripts

Capítulo 5

Desarrollo del software

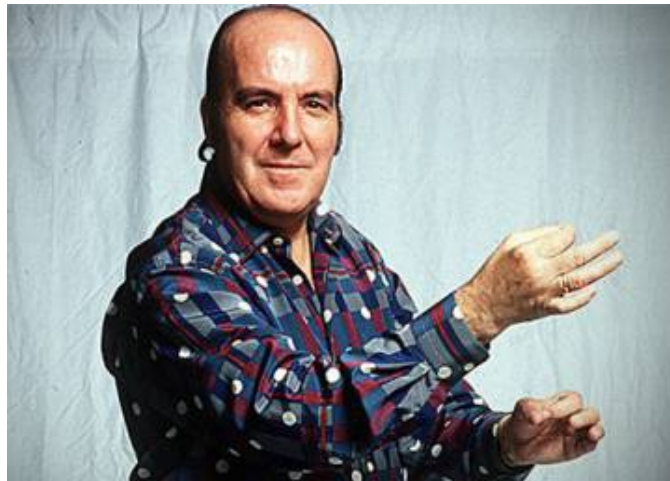


Figura 5.1: Raspberry, te das cuen

Capítulo 6

Pruebas

- 6.1. Comparación con otros procesadores de mayor potencia
- 6.2. Estudio a partir de qué nivel de uso del cluster merece la pena ante un procesador normal
- 6.3. Pruebas contra el cluster de Amazon(???)

Capítulo 7

Trabajo individual

7.1. Miguel Romero Martínez:

7.2. Daniel Quiñones Sánchez:

Capítulo 8

Conclusiones y trabajo futuro

8.1. Conclusiones generales

8.2. Trabajo futuro

8.3. General conclusions:

8.4. Future work:

Herramientas utilizadas

Github

Para realizar el trabajo de forma colaborativa, utilizamos un repositorio privado en la plataforma GitHub, lo que nos permite llevar un control de versiones del proyecto, dividirlo en ramas en función de la fase del desarrollo y mantenernos informados de los cambios que ha introducido cada miembro del grupo.

Latex

L^AT_EX es un sistema de preparación de documentos. Está orientado a la presentación de escritos que requieran de calidad profesional. Se compone de una serie de macros que ayudan a usar el lenguaje T_EX (Wikipedia, TeX). Permite, a su vez, separar el contenido del formato del documento. En este trabajo, hemos utilizado L^AT_EX para la maquetación de la memoria.

Bibliografía

- AB, M. Minecraft. 2011. Disponible en <https://minecraft.net/> (último acceso, Mayo, 2017).
- ALCALÁ, J. Inteligencia artificial en videojuegos. *Laboratorio de Investigación y Desarrollo en Inteligencia Artificial, Departamento de Ciencias e Ingeniería de la Computación, ????*
- ALGORITHMS y MORE. árbol de expansión mínima: Algoritmo de kruskal. 2012. Disponible en <https://jariasf.wordpress.com/2012/04/19/arbol-de-expansion-minima-algoritmo-de-kruskal/> (último acceso, Mayo, 2017).
- ARAUJO, L. y CERVIGÓN, C. *Algoritmos evolutivos: un enfoque práctico*. RA-MA S.A, 2009.
- BULLEN, T. y KATCHABAW, M. Using genetic algorithms to evolve character behaviours in modern video games problem encoding population initialization evaluation selection evolution population replacement. 2004.
- CAPCOM. Street fighter. 1987. Disponible en <http://www.streetfighter.com/> (último acceso, Mayo, 2017).
- CORPORATION, T. Space invaders. 1978. Disponible en <http://www.spaceinvaders.net/> (último acceso, Mayo, 2017).
- ECURED, P. Lisp. 2016. Disponible en <https://www.ecured.cu/Lisp> (último acceso, Mayo, 2017).
- MATEMÁTICA APLICADA Y ESTADÍSTICA, U. D. D. Test de turing. 2004. Disponible en <http://matap.dmae.upm.es/cienciaficcio/DIVULGACION/3/TestTuring.htm> (último acceso, Mayo, 2017).
- FONT, J. M., IZQUIERDO, R., MANRIQUE, D. y TOGELIUS, J. Constrained level generation through grammar-based evolutionary algorithms. 2016.

- GAMASUTRA. Procedural dungeon generation algorithm. 2014. Disponible en http://www.gamasutra.com/blogs/AAdonaac/20150903/252889/Procedural_Dungeon_Generation_Algorithm.php (último acceso, Mayo, 2017).
- GAMES, E. Unreal tournament. 1999. Disponible en <https://www.epicgames.com/unrealtournament/> (último acceso, Mayo, 2017).
- GAMES, H. No man's sky. 2016. Disponible en <https://www.nomanssky.com/> (último acceso, Mayo, 2017).
- GARCÍA-ORTEGA, R. y GARCÍA-SANCHEZ, P. My life as a sim: evolving unique and engaging life stories using virtual worlds. *ALIFE 14*, 2014.
- GUILLÉN TORRES, B. El verdadero padre de la inteligencia artificial. 2016. Disponible en <https://www.bbvaopenmind.com/el-verdadero-padre-de-la-inteligencia-artificial/> (último acceso, Mayo, 2017).
- INTRIAGO, J. Algoritmo a estrella. 2014. Disponible en <https://advanceintelligence.wordpress.com/2014/10/07/algoritmo-a-estrella/> (último acceso, Mayo, 2017).
- JACKSON, D. Evolving defence strategies by genetic programming. *EuroGP*, 2005.
- KOZA, J. R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.
- DEPARTAMENTO DE MATEMÁTICA APLICADA, U. P. D. M. Triangulación de delaunay. 2015. Disponible en http://www.dma.fi.upm.es/recursos/aplicaciones/geometria_computacional_y_grafos/web/triangulaciones/delaunay.html (último acceso, Mayo, 2017).
- MUÑOZ, M. Juegos roguelike: Historia y actualidad. 2014. Disponible en <http://www.fsgamer.com/juegos-roguelike-historia-y-actualidad-20140414.html> (último acceso, Mayo, 2017).
- NAMCO. Pacman. 1980. Disponible en <http://pacman.com/> (último acceso, Mayo, 2017).
- UNIVERSIDAD DE OVIEDO, C. D. I. A. Problema del coloreamiento de un grafo. 1997. Disponible en <http://www.aic.uniovi.es/ssii/Tutorial/Grafos.html> (último acceso, Mayo, 2017).
- PÉREZ, D., TOGELIUS, J., SAMOTHRAKIS, S., ROHLFSHAGEN, P. y LUCAS, S. Automated map generation for the physical travelling salesman problem. *Evolutionary Computation, IEEE Transactions on*, 2013.

-
- SOFTWARE, G. Borderlands. 2009. Disponible en <https://borderlandsthegame.com/> (último acceso, Mayo, 2017).
- TOGELIUS, J., PREUSS, M., BEUME, N., WESSING, S., HAGELBÄCK, J., YANNAKAKIS, G. N. y GRAPPIOLO, C. Controllable procedural map generation via multiobjective evolution. *Genetic Programming and Evolvable Machines*, 2013.
- TOY, M., WICHMAN, G. y ARNOLD, K. Rogue. 1980.
- UNITY. Angry bots. 2011.
- WIKIPEDIA (TeX). Entrada: “TeX”. Disponible en <https://es.wikipedia.org/wiki/TeX> (último acceso, Mayo, 2017).