

La plataforma Simics como herramienta de aprendizaje

Alberto Ros y José M. García

Departamento de Ingeniería y Tecnología de Computadores

Facultad de Informática. Universidad de Murcia

30100 Murcia

{a.ros,jmgarcia}@ditec.um.es

Resumen

En este trabajo mostramos como se imparte la asignatura *Diseño de arquitecturas de alto rendimiento* en la facultad de informática de la Universidad de Murcia. La principal innovación en la asignatura es el uso de la plataforma de simulación Simics como herramienta didáctica. En concreto en esta asignatura usamos Simics para la evaluación de servidores de aplicaciones. La experiencia en la asignatura ha sido muy positiva y los alumnos han respondido adecuadamente a la idea de tener que desarrollar un trabajo cercano al mundo empresarial, llegando a disfrutar con la evaluación y configuración de los equipos. Esto se demuestra en la valoración que ha obtenido la asignatura por los alumnos (8,11 sobre 10). Finalmente, hemos comprobado que al finalizar el curso los alumnos han mejorado sus conocimientos científicos y adquirido las competencias que se proponía la asignatura, tanto específicas como transversales.

1. Motivación

Con la llegada del Espacio Europeo de Educación Superior (EEES) se pretende que la enseñanza universitaria dote a los alumnos de una serie de competencias que les sirvan a la hora de enfrentarse al mercado laboral. La ANECA (Agencia Nacional de Evaluación de la Calidad y Acreditación) ha realizado un gran esfuerzo elaborando los denominados *libros blancos* en los que se recogen las com-

petencias que deben haber adquirido los estudiantes de cada titulación al acabar el grado. Dichas competencias han sido recopiladas mediante numerosos análisis, encuestas y reuniones con empresas, profesores y titulados, y clasificadas según dos tipos: las *específicas* de cada titulación y las *transversales* comunes a todas las titulaciones.

En este artículo pretendemos mostrar las modificaciones que hemos realizado en una asignatura del plan de estudios actual (*Diseño de arquitecturas de alto rendimiento*) para diseñarla teniendo en cuenta el análisis de competencias, y que nos sirva de experiencia para la inminente puesta en marcha de los nuevos planes de estudio de grado en informática.

Diseño de arquitecturas de alto rendimiento (DAAR) es una asignatura optativa de quinto curso de la ingeniería informática impartida en la Universidad de Murcia. Es una asignatura de carácter principalmente práctico en la que se enseña el uso de las técnicas de análisis y evaluación de prestaciones de una arquitectura por medio de un caso práctico y actual: un servidor de aplicaciones [10].

En esta asignatura pretendemos cubrir en profundidad dos de las competencias específicas recogidas en el libro blanco del título de grado en ingeniería informática [3]: *evaluación de requisitos hardware y metodologías de configuración*. En general, de entre las competencias específicas indicadas en el libro blanco, DAAR pretende dotar a los alumnos de las mostradas en la tabla 1. Además, en la asigna-

Tabla 1: Competencias específicas tratadas en DAAR

Competencias específicas
Arquitecturas de computadores
Diseño y arquitectura de sistemas de información
Evaluación de requisitos hardware
Metodologías de configuración
Métodos y Herramientas para el diseño y desarrollo de sistemas basados en computadores
Tecnología hardware
Visión comercial y empresarial

tura también se tratan las competencias transversales más importantes según el citado libro blanco como la *capacidad para resolver problemas*, el *trabajo en equipo* y la *capacidad de análisis y de síntesis*.

Al ser una asignatura optativa, el número de alumnos no es muy elevado, pero posee una tendencia ascendente. Por otro lado, el reducido número de alumnos permite un seguimiento cercano del trabajo realizado por ellos. Por esa razón se ha elegido un método de evaluación continua, en la que la nota de los alumnos depende de los progresos realizados.

En el curso 2006/2007, se propuso por primera vez en la asignatura el uso del simulador Simics para la evaluación de prestaciones de los servidores de aplicaciones que utilizamos para las prácticas. La ventaja principal de Simics consiste en que permite modificar la configuración de los servidores de un modo sencillo, pudiéndose evaluar diversas configuraciones. Gracias al uso de Simics se pueden modelar y evaluar diversas configuraciones de la arquitectura del simulador modificando parámetros que influirán en el rendimiento, como el número de procesadores, el tamaño de la memoria o la red de interconexión. De esta forma el alumno se enfrenta a decisiones de diseño, preguntándose si conviene mejorar un componente u otro para tener un mejor ratio rendimiento/coste.

Finalmente, se pretende que la asignatura sea muy útil, para que los alumnos puedan

aplicar estos conocimientos a los servidores que tienen las empresas de tamaño pequeño y mediano, que son el tipo de empresas más comunes dentro de nuestro entorno socioeconómico.

Tras esta introducción, el resto del artículo se estructura de la siguiente manera. En la sección 2 se comentará la importancia de aprender a evaluar y configurar los servidores de aplicaciones. La sección 3 estudia las herramientas de simulación disponibles y por que Simics es la más adecuada para la asignatura. La sección 4 introduce la plataforma de simulación Simics. A continuación, en la sección 5 mostramos la experiencia práctica del año pasado en el que incorporamos por primera vez el uso de Simics a la asignatura, ofreciendo en la sección siguiente algunos resultados acerca de este sistema, tanto a juicio de los alumnos como de los profesores. Finalmente, el artículo termina con una sección de conclusiones acerca de todo lo expuesto.

2. La evaluación de servidores de aplicaciones

Actualmente las empresas y los organismos públicos quieren ofrecer nuevos servicios a sus clientes y ciudadanos haciendo uso de las facilidades que provee Internet, es decir por medio de servicios basados en web. El responsable de estos sistemas informáticos necesita conocer cómo de eficiente es el sistema, si dicho sistema podría soportar la carga de trabajo del próximo año, qué componente o característica del sistema es necesario mejorar para poder soportar una carga mayor, etc. Para dar respuesta a estas cuestiones hay que utilizar técnicas cuantitativas que arrojen algo de luz a las preguntas realizadas [5].

La mejora del rendimiento de un sistema informático, por tanto, no es tarea trivial, ya que hay multitud de factores que influyen en él, desde los componentes físicos del computador hasta el comportamiento de los programas que se ejecutan en él (sistema operativo y aplicaciones de usuario). En cualquier caso, la mejora del rendimiento implicará la locali-

zación del cuello de botella del sistema para realizar las oportunas acciones que permitan mejorar dicho sistema.

En la asignatura que impartimos enseñamos a los alumnos a afrontar el reto de evaluar y mejorar el rendimiento de un servidor de altas prestaciones. De hecho, las prácticas que se desarrollan a lo largo de la asignatura se sitúan en esta línea, estando enmarcadas en el contexto definido por el rendimiento de un servidor con contenidos tanto estáticos como dinámicos. Sin embargo, a la hora de evaluar el rendimiento de un servidor no es posible modificar de un modo sencillo las características del sistema y, por tanto, sólo es posible detectar los cuellos de botella, sin conocer como el cambio de este componente mejoraría en el rendimiento del servidor.

El uso de simuladores facilita esta tarea permitiendo modificar parámetros tales como el número de procesadores, tamaño de las caches y memoria, red de interconexión, etc. Al contrario que las aplicaciones de tipo científico, las aplicaciones que se ejecutan en los servidores presentan una serie de dificultades para su estudio debido a la alta influencia de la actividad del sistema operativo, la red de comunicación y el sistema de almacenamiento. Por tanto, para su correcta evaluación es necesario el uso de un simulador que modele un sistema completo.

3. Herramientas de Simulación

Las herramientas de simulación ofrecen una serie de posibilidades que no están disponibles en una máquina real. Además de permitir modificar las características de la máquina, ofrecen estadísticas detalladas de la simulación. Por otro lado, al ser deterministas, se puede repetir sucesivas veces una simulación obteniendo siempre el mismo resultado. Sin embargo, el inconveniente de la simulación consiste en la velocidad. Obviamente ejecución de una aplicación en una máquina real, es más rápida que su simulación.

Existen muchos simuladores utilizados tanto en docencia como en investigación. Ejem-

plos ellos son *RSIM* [6], *SimpleScalar* [4], *GEMS* [9], etc. Estos simuladores permiten modificar diversos aspectos del sistema de un modo sencillo mediante parámetros. Sin embargo, estas herramientas no simulan el comportamiento de una máquina completa (por ejemplo, las llamadas al sistema), y por tanto no se adecuan a los objetivos de las prácticas. Además, la adaptación de las aplicaciones para simularlas bajo estas herramientas resultaría tediosa.

En el otro extremo se encuentran las herramientas de virtualización, en las que es posible instalar cualquier sistema operativo. En algunos casos, como en *VMWare* [12], sin necesidad de modificación. En otros, como ocurre en *Xen* [1], el sistema operativo requiere modificaciones para su instalación. Desafortunadamente, estas herramientas no permiten modificar la configuración de la máquina virtualizada, por lo que tampoco son adecuados para los objetivos de la asignatura.

En el termino medio se encuentra el simulador *Simics* [8]. *Simics* simula un sistema completo permitiendo la instalación de un sistema operativo sin modificaciones y, por tanto, la instalación de las aplicaciones que van a ser evaluadas. Además permite modificar la máquina simulada y ofrece estadísticas que ayudan a comprender el funcionamiento de la aplicación. Es por ello, que creemos que *Simics* es la herramienta que mejor se ajusta a los objetivos de la asignatura.

4. La plataforma Simics

Virtutech *Simics* [8] es una plataforma de simulación de un sistema completo que permite simular el conjunto de instrucciones de diversas arquitecturas. Se utiliza principalmente para tareas de investigación en arquitectura de computadores, desarrollo de sistemas operativos y simulación conjunta de hardware y software. En DAAR proponemos el uso de *Simics* con fines docentes, para la evaluación de servidores de aplicaciones.

A la hora de analizar el rendimiento de un servidor de aplicaciones, resulta relativamente

sencillo (mediante *benchmarking* o otras técnicas) obtener la máxima carga que puede soportar un sistema real para una aplicación determinada. Sin embargo, debido a la dificultad de modificar los componentes hardware de dicho sistema, resulta complicado averiguar como afecta el cambio de un componente al rendimiento del servidor.

Es por ello por lo que proponemos el uso de Simics como herramienta docente. Simics permite ejecutar cualquier sistema operativo sin modificaciones sobre una máquina con las características deseadas, permitiendo variar el número de procesadores, la frecuencia del procesador, el tamaño de la jerarquía de memoria, la velocidad de la red, etc. Esta flexibilidad convierte a Simics en una herramienta óptima para el estudio del rendimiento de servidores de aplicaciones, ya que permite experimentar de un modo sencillo con una gran variedad de máquinas y configuraciones.

Además, Simics se puede usar de un modo sencillo como herramienta para la docencia. Una vez conocidos una serie de comandos básicos (los cuales explicamos mediante seminarios), y arrancado el simulador, se dispone de una o varias terminales, cada una de ellas representando un equipo (servidor o clientes). A modo de ejemplo, en la figura 1 se pueden observar dos instancias de Simics conectadas mediante una conexión de red simulada. La máquina cliente (esquina inferior derecha) está realizando un *ping* sobre la máquina servidor (esquina superior izquierda). La tercera ventana (esquina inferior izquierda) corresponde a la consola de Simics y permite detener y reanudar la simulación en cualquier momento, así como poder guardar el estado completo de la máquina. Esto es muy útil para los alumnos que desean trabajar en los laboratorios disponibles para la asignatura, ya que pueden guardar todo el trabajo realizado al final de una sesión y cargarlo en cualquier otro momento, reanudándose la simulación por donde se detuvo la última vez.

Para la realización de las prácticas, los alumnos deben instalar la aplicación web en el servidor y un *benchmark* en el cliente, que

lance peticiones sobre el servidor. La instalación de la aplicación en el simulador se realiza del mismo modo que en un equipo real que contara con el mismo sistema operativo.

4.1. Primeros pasos en Simics

En esta sección comentaremos brevemente los pasos a seguir por los alumnos para la instalación y configuración de Simics, con el fin de demostrar su fácil manejo. Los recursos software necesarios para la instalación y uso del simulador se encuentran disponibles en todos los equipos de los laboratorios de prácticas. Estos recursos son los siguientes:

- Simulador Simics (*simics-pal-3.0.30-linux.tar*): Para la realización de las prácticas proporcionamos la versión 3.0 de Simics.
- Fichero de licencia (*u_de_murcia-x2008-08-31-singleserver.lic*): Para poder ejecutar Simics es necesario tener una licencia. Dicha licencia ha sido adquirida por la Universidad de Murcia. Aunque cada alumno puede obtener una licencia de estudiante para poder trabajar en casa, nosotros proporcionamos el fichero con la licencia de la Universidad de Murcia con el fin de evitar que el alumno tenga que registrarse.
- Imagen de disco (*tango1-fedora5.craff*): Para que Simics ejecute el sistema operativo al arrancar, se necesita un fichero que contiene una imagen del disco duro con el sistema operativo ya instalado. En concreto, proporcionamos la imagen de un disco con *fedora core 5*.

En primer lugar, el alumno debe instalar Simics y copiar el fichero de licencia. Los ficheros descritos anteriormente se encuentran a disposición de los alumnos en el directorio */tools/daar/*.



Figura 1: Dos equipos simulados por Simics conectados entre sí por un enlace de red simulado.

```
$ cp /tools/daar/simics-pal-3.0.30-linux.tar .
$ tar xf simics-pal-3.0.30-linux.tar
$ cd simics-3.0-install
$ sh install-simics.sh
$ cp /tools/daar/u_de_murcia-x2008-08-31-single
server.lic simics-3.0.30/licenses/
```

Después se ha de crear un directorio de trabajo (por ejemplo, *workspace*) y copiar en él el fichero con la imagen de disco.

```
$ cd simics-3.0.30
$ ./bin/workspace-setup workspace
$ cd workspace
$ cp /tools/daar/tango1-fedora5.craff .
```

Por último sólo quedaría ejecutar Simics. Por ejemplo, escribiendo el siguiente comando se ejecutarían dos instancias de Simics conectadas por un enlace simulado como el de la figura 1. Para que comience la ejecución de los sistemas hay que escribir el comando *c* en la consola de simics.

```
$ ./simics targets/x86-440bx/tango-multi.simics
simics> c
```

Al acabar de trabajar con Simics, se puede guardar el estado completo del sistema simulado mediante *checkpoints*. Posteriormente, el alumno podrá volver a cargar el sistema y continuar la simulación por donde la había dejado.

```
simics> # Guardar el checkpoint ...
simics> write-configuration checkpoint1
simics> # Cargar el checkpoint ...
simics> read-configuration checkpoint1
```

Simics se puede configurar con diversos parámetros y componentes. El sistema de configuración de Simics es orientado a objetos. Cada objeto es definido por una serie de propiedades llamadas *atributos*. Por ejemplo, el objeto *processor* tiene un atributo llamado *freq_mhz* que define su frecuencia. Sin embargo, Simics ofrece para cada arquitectura

simulada poder modificar una serie de parámetros de un modo más directo. Por ejemplo, escribiendo el código que se muestra a continuación en el fichero `.simics`, pasado como parámetro al simulador y que define la máquina simulada, ejecutaremos un sistema con dos procesadores de $1GHz$ cada uno con un total de $1GB$ de memoria.

```
$num_cpus = 2
$freq_mhz = 1000
$memory_megs = 1024
```

5. Experiencia práctica

En esta sección describimos como hemos llevado a la práctica el uso del simulador Simics con fines docentes en la asignatura *Diseño de arquitecturas de alto rendimiento*.

Como hemos comentado, DAAR es una asignatura de carácter primordialmente práctico. La idea de la asignatura es que los alumnos se acerquen al mundo empresarial mediante un caso práctico y actual, como es la evaluación y mejora de un servidor de aplicaciones.

Para potenciar el trabajo en equipo se anima a que los alumnos formen parejas desde el primer día de la asignatura. Cada equipo deberá elegir una aplicación a evaluar tanto en una máquina real como en una máquina simulada con Simics. La tabla 2 muestra las aplicaciones que pueden elegir los alumnos. Estas aplicaciones se pueden clasificar en estáticas y dinámicas. La elección de las estáticas implica un nivel de dificultad de las prácticas menor, ya que solo se precisa la instalación de la aplicación *apache* y el benchmark que genera la carga de trabajo [11]. En cambio, las aplicaciones dinámicas requieren además la instalación otras aplicaciones como *PHP* o *MySQL* en el caso de RUBiS y RUBBoS [2], o *PostgreSQL* en el caso de TPC-C [7].

Para la explicación de los contenidos teóricos necesarios para la elaboración de las prácticas utilizamos principalmente la clase magistral, mediante la transmisión de información en un tiempo ocupado principalmente por la exposición oral y el apoyo de las TICs. Du-

Tabla 2: Aplicaciones a elegir en las prácticas.

Estáticas	
Apache con httpperf	Servidor http
Apache con WebStone	Servidor http
Dinámicas	
SPECweb2005	Catálogo, Sitio de subastas, sitio de banca
SPECjbb2005	Servidor de una compañía de venta al por mayor
TPC-C	Servidor de procesamiento de transacciones en línea
RUBiS	Sitio de subastas
RUBBoS	Foro de noticias

rante dicha exposición planteamos preguntas, resolvemos las dudas, orientamos la búsqueda de información, y fomentamos el debate individual o en grupo.

Una vez explicados los contenidos teóricos, los alumnos proceden a la elaboración de las prácticas, estructuradas en tres bloques. Cada bloque irá seguido de una defensa del trabajo realizado, para fomentar el desarrollo de las habilidades de comunicación oral por parte de los alumnos. La evaluación de las tres prácticas y las respectivas defensas se hace de forma continua. Tras la defensa de cada uno de los trabajos, tanto los alumnos como los profesores podrán formular preguntas a los compañeros que defienden su trabajo. Al final, los profesores comentan los puntos fuertes y débiles de los trabajos con el fin de que sirvan de *feedback* para la realización de las practicas siguientes.

El la primera práctica los alumnos deberán instalar y configurar la aplicación elegida en una máquina real. Una vez instalada, los alumnos deben realizar la evaluación de las prestaciones del servidor, obteniendo su capacidad máxima de carga y el tiempo mínimo de respuesta y localizando su cuello de botella.

La segunda práctica consistirá en realizar los mismos pasos sobre Simics. Los alumnos deben analizar los resultados obtenidos en la

evaluación del servidor y compararlos con los obtenidos en la primera práctica. Con esto pretendemos fomentar la capacidad de análisis y síntesis de los alumnos.

Finalmente, la tercera práctica consiste en usar la capacidad de Simics para modificar la configuración del servidor de un modo inteligente. Los alumnos deben buscar una configuración lo más económica posible que mejore el rendimiento del sistema.

6. Resultados obtenidos

Con la intención de conocer el punto de vista de los alumnos sobre la asignatura, realizamos al final del curso una encuesta. La encuesta la realizamos con la aplicación web *PHPSurveyor*¹. Esta aplicación proporciona dos ventajas respecto a las encuestas tradicionales. En primer lugar, realizar la encuesta se convierte en una tarea más amena para los alumnos. En segundo lugar, los resultados se almacenan en una base de datos que se puede manejar con facilidad para obtener los datos necesarios.

Desde nuestro punto de vista, la experiencia en la asignatura ha sido muy positiva. Hemos notado que los alumnos responden adecuadamente a la idea de tener que desarrollar un trabajo cercano al mundo empresarial, llegando algunos de ellos a disfrutar con la evaluación y configuración de los equipos. Pero sin lugar a dudas lo más importante es que los alumnos mejoran sus conocimientos sobre arquitectura de computadores, configuración de sistemas, evaluación de prestaciones y aplicaciones web.

La inclusión del simulador Simics en la asignatura ha supuesto también un avance en la enseñanza de los alumnos. La posibilidad de poder evaluar una gran variedad de configuraciones y equipos para una aplicación web no era posible antes de la inclusión de este simulador. En nuestro primer año realizando prácticas con Simics, todos los alumnos que realizaron las tres presentaciones de las prácticas aprobaron la asignatura. Además notamos un

¹ *PHPSurveyor* está disponible en la dirección web <http://sourceforge.net/projects/phpsurvey/>

aumento de la calidad de las presentaciones y su contenido conforme avanzaba la asignatura.

Con respecto a la opinión de los alumnos sobre la asignatura, y a partir de los datos recogidos en la encuesta, todos los alumnos piensan que el grado de complejidad de Simics es adecuado para quinto curso, aunque el 11% piensa que las prácticas realizadas no reflejan adecuadamente los contenidos de la asignatura. Por último, los alumnos piensan que la asignatura es útil para su formación profesional y científica (4,33 sobre 5), y de hecho la recomendarían a otros compañeros (4,56 sobre 5). Al final del cuestionario preguntamos a los alumnos que evaluarán la asignatura con una puntuación del 1 al 10, obteniendo un resultado de 8,11.

7. Conclusiones

En este trabajo hemos mostrado como usar el simulador Simics con fines docentes en informática, en concreto para la evaluación de servidores de aplicaciones. El caso de uso elegido ha sido la asignatura optativa de quinto curso *Diseño de arquitecturas de alto rendimiento* de la facultad de informática de la Universidad de Murcia.

La innovación presentada se sitúa en el marco de desarrollo de asignaturas teniendo en cuenta las directivas marcadas por el EEES. Concretamente, hemos realizado el programa de la asignatura en base a las competencias a alcanzar, y el desarrollo de la misma nos ha permitido interactuar y tutorizar a los alumnos de forma adecuada.

La experiencia en la asignatura ha sido positiva. Hemos notado que los alumnos responden adecuadamente a la idea de tener que desarrollar un trabajo cercano al mundo empresarial, llegando a disfrutar con la evaluación y configuración de los equipos. Pero sin lugar a dudas lo más importante es que los alumnos mejoran su conocimiento sobre arquitectura de computadores, configuración de sistemas, evaluación de prestaciones y aplicaciones web. Las prestaciones que ofrece Simics para modi-

ficar la configuración de los equipos ha sido fundamental para lograr estos objetivos.

Los alumnos han respondido bien a la asignatura y han quedado bastante satisfechos, asignando a la asignatura una puntuación de 8,11 sobre 10. Además, la mayoría cree que Simics es adecuado para la asignatura tanto en términos de complejidad como de contenido. En general, piensan que la asignatura es de gran utilidad para su formación profesional y científica.

Finalmente, hemos comprobado que al cursar la asignatura los alumnos adquieren las competencias propuestas, tanto transversales como específicas.

Agradecimientos

Este trabajo ha sido financiado por el Ministerio de Educación y Ciencia a través del proyecto "TIN2006-15516-C04-03". Alberto Ros agradece al Ministerio de Educación y Ciencia la financiación ofrecida dentro del programa FPU, con la beca AP-2004-3735.

Referencias

- [1] Tim Abels, Puneet Dhawan y Balasubramanian Chandrasekaran, *An overview of xen virtualization*, Dell Power Solutions, agosto 2005.
- [2] Cristiana Amza, Emmanuel Cecchet, Anupam Chanda, Alan Cox, Sameh Elnikety, Romer Gil, Julie Marguerite, Kartick Rajamani y Willy Zwaenepoel, *Specification and Implementation of Dynamic Web Site Benchmarks*, WWC-5: IEEE 5th Annual Workshop on Workload Characterization, noviembre 2002.
- [3] ANECA, *Libro Blanco del Título de Grado en Ingeniería Informática*, Proyecto EICE, junio 2005.
- [4] Todd Austin, Eric Larson y Dan Ernst, *SimpleScalar: An Infrastructure for Computer System Modeling* IEEE Computer, 35(2), pp. 59-67, febrero 2002.
- [5] John L. Hennessy y David A. Patterson, *Computer Architecture: A Quantitative Approach*, 4a Edición, Morgan Kaufmann Publishers, Inc., 2007.
- [6] Christopher J. Hughes, Vijay S. Pai, Parthasarathan Ranganathan y Sarita V. Adve, *RSIM: Simulating Shared-Memory Multiprocessors with ILP Processors*, IEEE Computer, 35(2), pp. 40-49, febrero 2002.
- [7] Diego R. Llanos, *TPCC-UVA: An Open-Source TPC-C Implementation for Global Performance Measurement of Computer Systems*, ACM SIGMOD Record, ACM Press, diciembre 2006.
- [8] Peter S. Magnusson, Magnus Christenson, Jesper Eskilson, Daniel Forsgren, Gustav Hallberg, Johan Hogberg, Fredrik Larsson, Andreas Moestedt y Bengt Werner, *Simics: A Full System Simulation Platform*, IEEE Computer, pp. 50-58, vol. 35, febrero 2002.
- [9] Milo M.K. Martin, Daniel J. Sorin, Bradford M. Beckmann, Michael R. Marty, Min Xu, Alaa R. Alameldeen, Kevin E. Moore, Mark D. Hill y David A. Wood *Multifacet's General Execution-Driven Multiprocessor Simulator GEMS Toolset*, Computer Architecture News, pp. 92-99, septiembre 2005.
- [10] Xavier Molero, Carlos Juiz y Miguel Rodero, *Evaluación y Modelado del Rendimiento de los Sistemas Informáticos*, Prentice Hall, 2004.
- [11] David Mosberger y Tai Jin, *httpperf: A Tool for Measuring Web Server Performance*, Internet Server Performance Workshop, pp. 59-67, junio 1998.
- [12] Jeremy Sugerman, Ganesh Venkitachalam y Beng-Hong Lim, *Virtualizing I/O devices on VMware workstation's hosted virtual machine monitor*, Annual Usenix Technical Conference, junio 2001.