

Integración del comportamiento dinámico del usuario en TPC-W¹

Raúl Peña-Ortiz, José Antonio Gil, Julio Sahuquillo, Ana Pont²

*Departament d' Informàtica de Sistemes i Computadors (DISCA)
Universitat Politècnica de València
Valencia, Spain*

Resumen

En los últimos años venimos asistiendo a un aumento en la cantidad de servicios ofrecidos a través de la World Wide Web (Web). Estos servicios han ido evolucionando paulatinamente, desde los primitivos servicios estáticos de primera generación, hasta las complejas y personalizadas aplicaciones web actuales, en las que el usuario es algo más que un mero espectador y se ha convertido en un creador de contenido dinámico. Esta evolución ha producido a su vez una evolución en las pautas de comportamiento de estos usuarios, que resultan cada vez más dinámicas. Consecuencia directa de la evolución de la Web es la necesidad de nuevas herramientas para una evaluación de prestaciones más acorde con las características dinámicas de la misma; herramientas que deben ser capaces de representar el comportamiento dinámico del usuario en la generación de la carga web.

Este trabajo presenta un nuevo entorno de prueba capaz de incorporar generación de carga dinámica en la evaluación de prestaciones de sistemas de comercio electrónico basados en web. Con tal fin, se ha partido del reconocido *benchmark* de comercio electrónico TPC-W y se ha integrado la generación de carga dinámica proporcionada por el generador GUERNICA, aprovechando sus cualidades a la hora de caracterizar y reproducir carga web basada en los patrones dinámicos del comportamiento del usuario. El nuevo entorno ha sido validado contra TPC-W, mostrando resultados similares cuando no se considera dinamismo en la caracterización de la carga.

Keywords: Evaluación de prestaciones web, generador de carga web, carga web dinámica, modelado del comportamiento dinámico del usuario.

¹ Este trabajo ha sido parcialmente financiado por el Ministerio de Ciencia e Innovación en el proyecto TIN2009-08201

² Emails: rpenya@upvnet.upv.es, {jagil, jsahuqui, apont}@disca.upv.es

1. Introducción

Los servicios ofertados a través de la World Wide Web (Web) han sufrido una constante evolución en los últimos años debido a los incesantes cambios de la tecnología web, lo que ha propiciado la aparición de nuevos tipos de comportamiento en los usuarios de la Web [18].

En los primitivos servicios estáticos de la primera generación, la Web suponía un medio de bajo coste para compartir información de escasa o nula privacidad, y la información era principalmente de tipo texto, con un pequeño porcentaje de imágenes embebidas. El usuario tipo de esta Web era un mero espectador que se limitaba a consultar información y navegar de acuerdo a los enlaces que encontraba en las páginas visitadas [8]. Posteriormente, los contenidos dinámicos alcanzaron gran auge, dando lugar a la segunda generación de servicios basados en web. Esta generación se caracterizó por fuertes cambios en sus infraestructuras y arquitecturas (p.e., implantación de sistemas de información basados en web, soportados por servidores de aplicaciones y bases de datos) que permitieron la generación, interrogación y almacenamiento dinámicos de información; dinamismo que se extendió al comportamiento del usuario [15], a sus pautas de navegación (más dinámicas y personalizadas), y por lo tanto al tráfico generado por las mismas [2]. En la actualidad, nos encontramos con una nueva oferta servicios donde el usuario ha dejado de ser únicamente consumidor de información, para pasar a participar activamente en la creación de contenidos personalizados y en la difusión o recomendación de los mismos [8].

Como todo sistema en continuo cambio, tanto en sus aplicaciones como en las infraestructuras que las sustentan, los estudios de evaluación de prestaciones son pieza clave para presentar propuestas apropiadas cuando se diseñan nuevos sistemas web [5] (p.e., servicios web, servidores web, proxies o políticas de distribución de contenidos). Todo proceso de evaluación de prestaciones debe emplear modelos de carga precisos y representativos para garantizar la validez de los resultados. En el caso de la Web, el dinamismo implícito en el comportamiento de sus usuarios dificulta el diseño de modelos capaces de representar las navegaciones reales.

En trabajos previos [17] introdujimos un nuevo modelo para caracterizar carga web dinámica, denominado Dweb. Este modelo está basado en el comportamiento dinámico del usuario y permite representar su capacidad para cambiar su comportamiento a lo largo del tiempo, adoptando dinámicamente diferentes roles (p.e., surfer, buscador o comprador) y por lo tanto navegando la Web de diferentes maneras. El generador de carga GUERNICA fue implementando tomando como base Dweb.

En el presente trabajo proponemos un nuevo benchmark web con capacidad para generar carga dinámica. Este nuevo entorno toma como base el reconocido

benchmark TPC-W en el que se ha integrado el proceso de generación de carga de GUERNICA, con el fin de dotarle de la capacidad de caracterizar y generar carga web dinámica a través de Dweb.

El resto del artículo se estructura como sigue. La sección 2 discute las razones que nos han llevado a realizar esta nueva propuesta de benchmark web. Las secciones 3 y 4 presentan y validan nuestra propuesta, respectivamente. Finalmente, exponemos algunas conclusiones finales y trabajo futuro en la sección 5.

2. Motivación y trabajo relacionado

La necesidad de modelos de carga basados en el comportamiento del usuario [4] aparece con la creciente importancia de las aplicaciones web. Esta necesidad es especialmente relevante en los entornos de comercio electrónico, donde la caracterización del comportamiento del usuario no es sólo objetivo de la evaluación de prestaciones web, sino que también juega un importante papel en términos de fidelización de clientes. En este tipo de aplicaciones se pone de manifiesto, entre otras, las siguientes características: i) importancia de la información crítica; ii) elevado porcentaje de contenido dinámico y personalizado; iii) necesidad en la calidad del servicio y calidad del producto que se ofrece a los usuarios que son tratados como clientes potenciales; y iv) incorporación de tecnología de última generación. En consecuencia, la utilización de modelos imprecisos en la evaluación de aplicaciones de comercio electrónico puede derivar en conclusiones incorrectas que supongan acciones inapropiadas sobre las prestaciones del sistema y sobre el desarrollo de negocio.

Floyd et al. [11] describen los inconvenientes de evaluar las prestaciones web mediante modelos de carga analíticos, debido principalmente al componente dinámico de la carga y a la gran diversidad de parámetros que influyen en la caracterización de los modelos analíticos (e.g., diferentes protocolos, tipos característicos de tráfico, patrones de navegación en los usuarios, etc). En general, los retos actuales en la caracterización de la carga web son: i) el modelado del comportamiento dinámico del usuario [4], ii) la definición de los roles que el usuario juega cuando navega la web [20] y iii) la representación de los continuos cambios de rol [13].

Existen pocos aunque interesantes esfuerzos para definir el comportamiento del usuario con el fin de caracterizar carga web representativa de cierto tipo de aplicaciones. Menascé et al. [14] introdujeron el Customer Behavior Model Graph (CBMG) que describe patrones de comportamiento del usuario en la carga relativa a aplicaciones de comercio electrónico. Duarte et al. [10] aplicaron este modelo para definir la carga de la blogsfera; Shams et al. [19] extendieron CBMG para reflejar las dependencias existentes entre las peticiones HTTP de una navegación y entre los datos relativos al contexto de

la misma. Benevenuto et al. [6] introdujeron el modelo Clickstream para caracterizar el comportamiento del usuario en las redes sociales. Sin embargo, estos modelos sólo caracterizan la carga web en paradigmas o aplicaciones de propósito específico, y no abordan el segundo y tercero de los retos mencionados anteriormente. Estas deficiencias nos motivaron a proponer un modelo de carga de propósito general denominado Dweb [17], que nos permite considerar modelos del comportamiento dinámico del usuario en la caracterización de la carga. Dweb representa el dinamismo del usuario de forma precisa teniendo en cuenta los tres retos mencionados.

Los estudios de evaluación de prestaciones web son soportados por software dedicado, que tiene el objetivo de validar la calidad de servicio de un sistema bajo condiciones de carga específicas definidas normalmente por modelos de carga. Existen varios tipos de software dedicado a la evaluación de prestaciones, de entre los cuales podemos destacar los benchmarks y los generadores de carga. Los primeros persiguen reproducir las condiciones de carga típicas del entorno de trabajo habitual, con el fin de constatar si el sistema evaluado cumple con las pautas de calidad establecidas. Los segundos buscan la generación de un número de peticiones HTTP lo suficientemente importante como para conseguir una degradación significativa de la calidad del servicio, que podría llegar a la denegación del mismo. De entre todas las herramientas software evaluadas en un trabajo previo [16], el benchmark TPC-W es el mejor entorno de pruebas representativo de un sistema de comercio electrónico, mientras que GUERNICA es el único generador que reproduce de forma precisa la carga web dinámica, a través del uso que hace de Dweb. TPC-W reproduce múltiples sesiones concurrentes de clientes sobre una librería on-line, pero no genera carga dinámica precisa porque sólo incluye una representación parcial del comportamiento dinámico del usuario basada en el uso del modelo CBMG.

En consecuencia, en el presente trabajo proponemos un nuevo benchmark para entornos de comercio electrónico con capacidad para caracterizar y generar carga web dinámica. Este benchmark se diseña como una extensión de TPC-W en cuya arquitectura se introduce la generación de carga dinámica basada en Dweb mediante la integración del núcleo de GUERNICA.

3. Integración entre TPC-W y GUERNICA

El desarrollo del benchmark ideado se ha realizado de acuerdo a tres premisas. En primer lugar, el benchmark debe definir y reproducir de forma apropiada y precisa la carga dinámica. En segundo lugar, debemos obtener un entorno que facilite las medidas de aquellas métricas de rendimiento, tanto en la parte cliente como en la parte servidor, imprescindibles en los estudios de evaluación de prestaciones cuando contemplamos diferentes cargas web. Fi-

nalmente, el benchmark debe proporcionar un entorno representativo de los sistemas web transaccionales que se han establecido en los últimos años.

De entre todas las alternativas de generación de carga web evaluadas en trabajos anteriores [16], TPC-W es el único benchmark que se ciñe a la mayoría de las premisas anteriores, pero aunque contempla el comportamiento dinámico del usuario en la caracterización de la carga, no lo hace de forma precisa. Consecuentemente, proponemos una extensión de TPC-W en la que se contempla el uso de GUERNICA para generar carga web dinámica de forma apropiada, lo que nos permite cumplir con las tres premisas.

La sección 3.1 presenta las principales características de TPC-W y de la implementación seleccionada como base de nuestro desarrollo. Posteriormente, presentamos las principales funcionalidades de GUERNICA y su arquitectura básica en la sección 3.2. Finalmente, la sección 3.3 introduce la arquitectura de integración entre TPC-W y GUERNICA que nos habilita a contemplar de forma apropiada y precisa carga dinámica en los estudios en los que emplear el benchmark.

3.1. TPC BenchmarkTM W

TPC BenchmarkTM W (TPC-W) es un benchmark de web transaccional que simula las principales actividades de un sitio web de comercio electrónico, concretamente de una tienda de libros on-line [1]. El benchmark reproduce la carga generada por múltiples sesiones concurrentes de clientes sobre una aplicación web que se encarga de servir los contenidos estáticos y dinámicos asociados a las actividades de consulta y venta de la tienda.

TPC-W proporciona un entorno estándar, independiente de la tecnología de implementación, de la arquitectura y de la infraestructura, que ha sido altamente contrastado y aceptado por la comunidad científico-técnica en numerosos estudios de evaluación de prestaciones web [9], [3], [12]. Como todo benchmark de comercio electrónico, TPC-W presenta una arquitectura cliente-servidor, recogida en la Figura 1. Los agentes software ubicados en la parte cliente (Remote Browser Emulators) son los encargados de generar carga sobre la aplicación de comercio electrónico del servidor (E-commerce server). Con el fin de reproducir una carga web representativa, los agentes simulan el comportamiento que debería tener un usuario real al navegar por el sitio web. El servidor alberga el sistema bajo prueba (Server Under Test), que se compone de: i) un servidor web y su sistema de almacenamiento de objetos estáticos, y ii) un servidor de aplicaciones y una base de datos para la generación de contenido dinámico. La pasarela de pagos (Payment Gateway Emulator) representa la entidad encargada de autenticar a los usuarios y autorizar sus pagos. Los tres componentes principales de la arquitectura comunican entre sí a través de una red de interconexión dedicada.

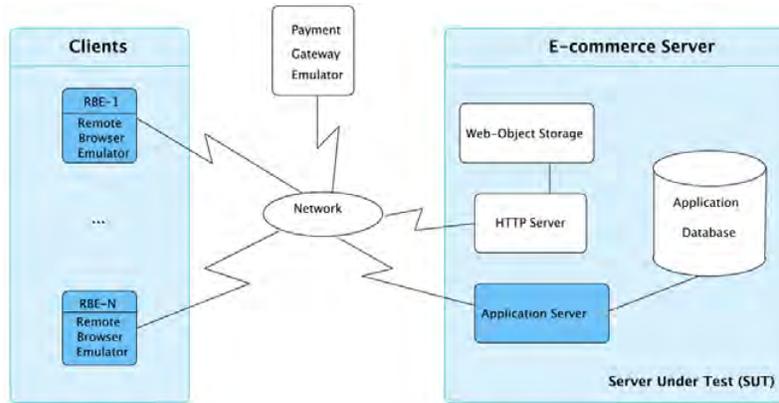


Figura 1. Arquitectura TPC-W

Para nuestros propósitos, adoptamos una implementación Java de TPC-W realizada por el Computer Architecture Group de la UW-Madison [7]. Como muestra la Figura 2, la parte cliente de su arquitectura está concebida como una aplicación de consola y proporciona dos interfaces relacionadas con el proceso de generación de carga: i) el agente software encargado de simular a los clientes (EB), y ii) la factoría de agentes (EBFactory), cuya función es la de crear y configurar los agentes proporcionados. Estas interfaces actúan como punto de extensión y permiten personalizar la caracterización de la carga web. La parte del servidor se ha desarrollado como una aplicación web compuesta por un conjunto de **Servlets** que se encargan de recoger las peticiones de los clientes y acceder a la base de datos para poder servirlos.

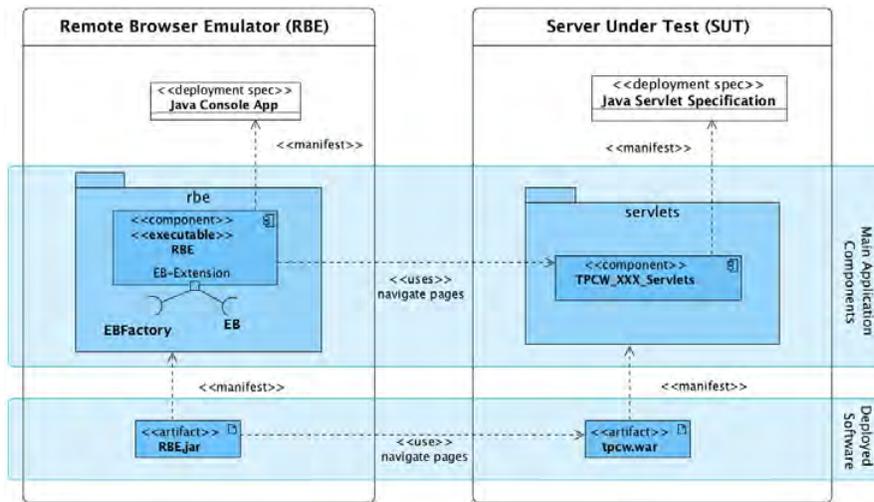


Figura 2. Componentes principales en la implementación Java de TPC-W

3.2. GUERNICA

GUERNICA (Universal Generator of Dynamic Workload under WWW Platforms) es un software generador de carga web resultado de la cooperación del Grupo de Investigación en Arquitectura y Prestaciones de la Web de la Universitat Politècnica de València y la empresa Intelligent Software Components.

La principal característica de GUERNICA es el uso que hace de los conceptos del modelo Dweb (Dynamic web workload model) [17] a la hora de definir el comportamiento del usuario, lo que le permite resolver completamente los tres retos planteados en la caracterización de la carga dinámica. El concepto de navegación define el comportamiento del usuario mientras interactúa con la web y facilita la caracterización del dinamismo del usuario en sus navegaciones. Por otro lado el concepto de test de carga se asocia a un conjunto de navegaciones que definen los posibles comportamientos de un usuario y proporciona un mecanismo para modelar estos comportamientos y los posibles cambios contemplados.

Tal y como se ilustra en la Figura 3, GUERNICA se presenta como un conjunto de tres aplicaciones principales: generador de carga (workload generator client), evaluador de rendimiento (performance evaluator client) y planificador de los test de prestaciones (performance tests planner), que permiten independizar y distribuir en diferentes nodos las principales actividades de los procesos de evaluación de prestaciones y evaluación funcional de una aplicación web.

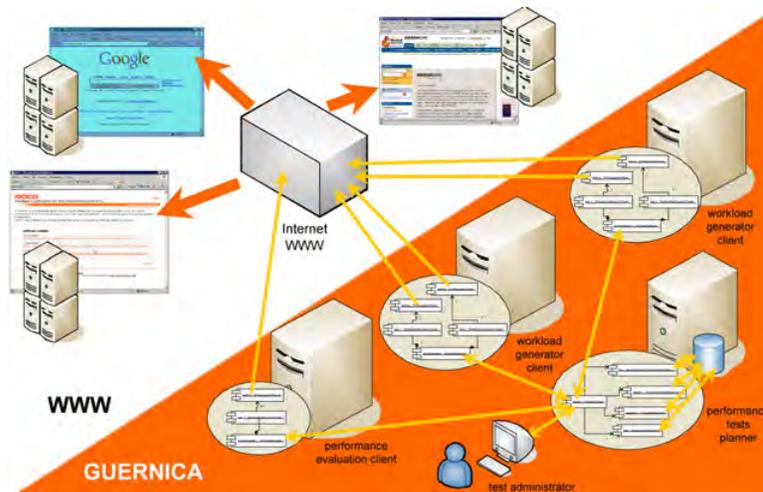


Figura 3. Principales aplicaciones en GUERNICA

Estas tres aplicaciones se definen dentro de una arquitectura basada en componentes, recogida en la Figura 4. El elemento central de la arquitectura, *GUERNICA.core*, es el encargado de implementar el proceso de generación de carga basado en Dweb. Los conceptos de test de carga y nave-

gación están representados respectivamente por las interfaces `WorkloadTest` y `WorkloadNavigation`. El componente encargado de simular el comportamiento de los usuarios recibe el nombre de `NavigationEngine`; su configuración se expresa en términos de los conceptos de Dweb y se almacena en un repositorio de nombre `WorkloadTestRespository`. El acceso centralizado a `GUERNICA.core` se lleva a cabo a través del `CoreManager`.

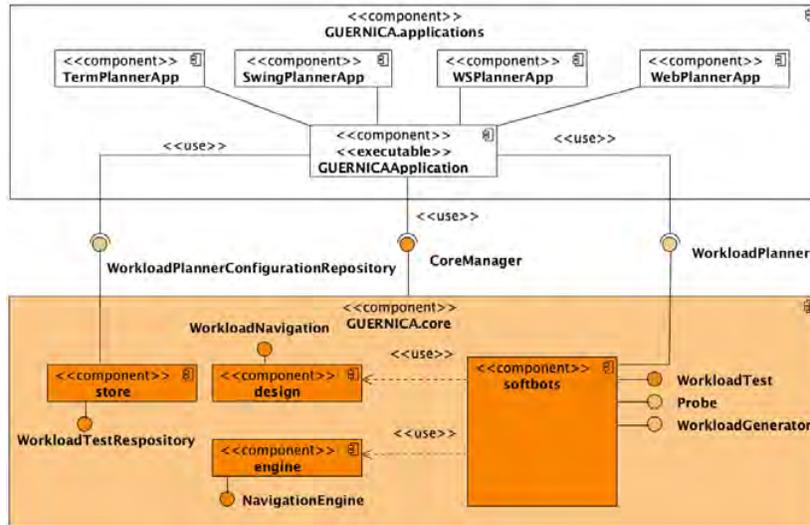


Figura 4. Principales componentes en GUERNICA

3.3. Arquitectura de integración

La arquitectura de integración entre TPC-W y GUERNICA (TGI) se recoge en la Figura 5. Dicha arquitectura se organiza en tres capas:

- La capa superior está definida por la parte cliente de TPC-W, que proporciona las dos interfaces principales del proceso de generación de carga (EB y `EBFactory`), detalladas en secciones previas.
- La capa inferior está relacionada con el proceso de generación de carga en GUERNICA, introducido en la sección anterior.
- Finalmente, la capa intermedia define la integración entre TPC-W y GUERNICA, la cual es suministrada como una librería Java independiente de TPC-W y de nombre TGI. Esta librería implementa un nuevo tipo de agente generador de carga (i.e., `DwebEB`) que usa el núcleo de GUERNICA para reproducir el comportamiento dinámico de los usuarios en el proceso de generación de carga. A fin de simplificar la implementación de este agente, un nuevo motor de carga (i.e., `DwebExecutorEngine`) ha sido implementado para llevar a cabo el proceso de generación. Adicionalmente, una nueva factoría de agentes (i.e., `DwebEBFactory`) se ha desarrollado para controlar

la configuración, creación y gestión de las instancias del nuevo agente en el entorno de TPC-W.

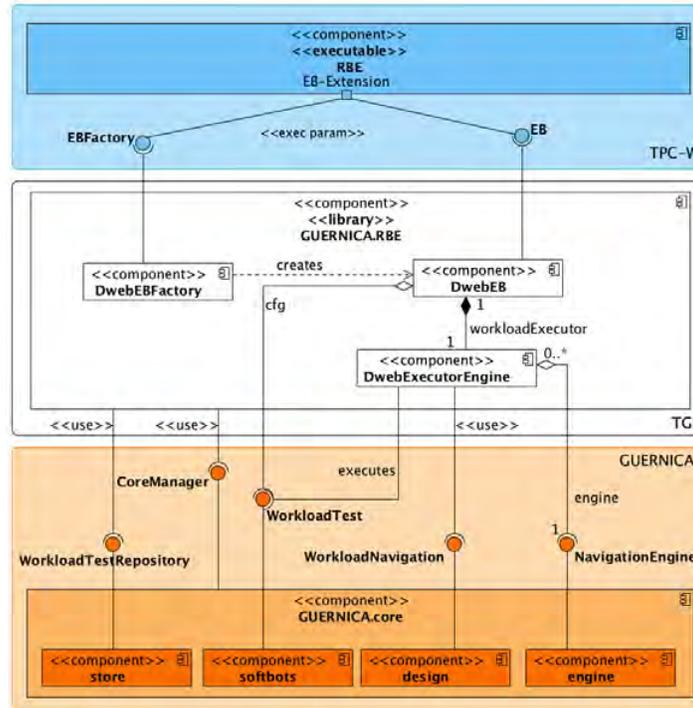


Figura 5. Arquitectura de TGI

4. Validación de TGI

Para poder explotar con garantías en trabajos futuros el nuevo benchmark, lo hemos validado contra TPC-W, considerando una configuración específica de entorno experimental para la cual se han medido sus principales métricas de rendimiento. Las Secciones 4.1 y 4.2 describen el entorno experimental y las métricas de rendimiento, respectivamente. La validación se detalla en la Sección 4.3.

4.1. Entorno experimental

La configuración utilizada en el entorno experimental para llevar a cabo la validación, sigue las pautas de una arquitectura tradicional cliente-servidor, que en nuestro caso consiste en un Servidor Ubuntu Linux como back-end y un Cliente Ubuntu Linux como capa de front-end. El back-end ejecuta la aplicación servidor de TPC-W, cuya base es una aplicación web Java (TPC-W web

app) que está desplegada en el servidor de aplicaciones Tomcat. Las peticiones a contenido estático, es decir las imágenes, son servidas directamente por el servidor web Apache. Las peticiones a contenido dinámico son redirigidas a Tomcat. La aplicación web se encarga de servir estas peticiones dinámicas accediendo a la información albergada en la base de datos MySQL. Por otro lado, en el front-end se ejecuta la aplicación del benchmark asociada a la generación de carga. Tanto la aplicación de back-end como la de front-end se ejecutan sobre el entorno Java SUN 5.0 (JRE 5.0). La Figura 6 ilustra el entorno experimental y detalla el hardware y software empleados.

Dada la arquitectura en capas del entorno experimental, el ajuste de los diferentes parámetros del sistema, tanto del servidor como de los generadores de carga, ha supuesto un punto fundamental para garantizar que los posibles cuellos de botella de la infraestructura empleada no distorsionan la validación realizada. Se ha empleado una configuración de TPC-W que contempla gran cantidad de artículos y clientes potenciales, concretamente 100.000 artículos y hasta 100 clientes potenciales registrados al seguir las reglas de escalabilidad del benchmark. Estas condiciones experimentales nos han obligado a revisar el ajuste del acceso a la base de datos (p.e., el tamaño del pool de conexiones), del servicio de contenido estático por parte de Apache (p.e., el número de workers que sirven peticiones HTTP), o del servicio de contenido dinámico por parte de Tomcat (p.e., el número de hilos para proporcionar contenido dinámico). En la validación del entorno, para cada tipo de carga de trabajo hemos realizado medidas de las métricas de rendimiento durante varias repeticiones de los experimentos, con el fin de recopilar sus valores de una manera precisa y representativa. Cada una de estas ejecuciones se compone de una fase de calentamiento de 15 minutos seguida de una fase de medición de 30 minutos.

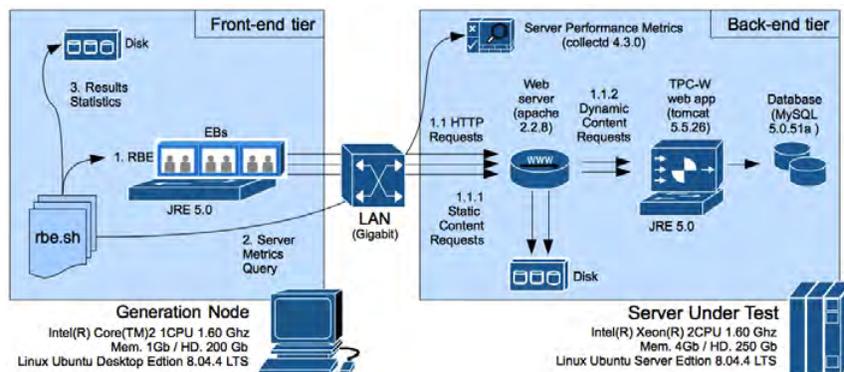


Figura 6. Entorno experimental

Tabla 1
Métricas de rendimiento

Recurso		Métrica	Descripción	
Cliente		WIRT	Web Interaction Response Time (WIRT) se define en TPC-W como $t2 - t1$, donde $t1$ es el tiempo medido en el agente generador de carga antes de enviar al servidor el primer byte de la primera petición HTTP en la interacción web, y $t2$ es el tiempo medido en el agente generador de carga después de recibir el último byte de la respuesta a la última petición HTTP que completa la interacción web con el servidor.	
		Req_{page}	Requests per Page (Req_{page}) es el número total de peticiones por página que son servidas con éxito.	
		\overline{WIRT}	$\overline{WIRT} = \frac{\sum_{i \in Pages} WIRT_i * Req_i}{\sum_{i \in Pages} Req_i}$.	
Servidor	Hardware	CPU	U_{CPU}	Las métricas para los principales recursos hardware incluyen la utilización para todos ellos, y la productividad para el disco y la red de interconexión.
		Memoria	U_{memory}	
		Disco	U_{disk}, X_{disk}	
		Red Interconexión	U_{net}, X_{net}	
	Software	Apache	$\{X, CPU, MEM\}_{apache}$	Los detalles de rendimiento de los principales componentes software del servidor incluyen: productividad, uso de la memoria y de la CPU por parte del componente, procesos o hilos de ejecución del componente, etc.
		Tomcat	$\{X, CPU, MEM\}_{tomcat}$	
		MySQL	$\{X, CPU, MEM\}_{mysql}$	

4.2. Métricas de rendimiento

La Tabla 1 resume aquellas métricas de rendimiento medidas en nuestro estudio. Las principales métricas consideradas desde el punto de vista del cliente son el tiempo de respuesta (WIRT) y el total de peticiones por página. En el servidor, nuestro estudio recoge las estadísticas del servidor que son requeridas por la especificación de TPC-W (p.e., utilización de la CPU, actividad de la E/S de la base de datos, actividad de la E/S del sistema o las estadísticas el servidor web) y además añade otras estadísticas opcionales. Las métricas recogidas se organizan en dos grupos de acuerdo a su naturaleza: i) métricas de los principales recursos hardware del sistema, y ii) detalles del rendimiento de los principales componentes software del back-end. Con el fin de estandarizar el proceso de evaluación de prestaciones empleamos un middleware llamado `collectd`³ que nos permite realizar una medida periódica.

4.3. Resultados experimentales

La validación del benchmark, requerida previamente a su explotación con garantías en estudios de evaluación de prestaciones, se ha realizado contra TPC-W con el fin de contrastar las principales funcionalidades y comportamiento de ambas implementaciones. Con este fin, identificamos un sitio web de validación a partir del mapa web de la librería on-line asociada a TPC-W.

Según la especificación de TPC-W, el CBMG completo para la librería on-line se compone de 14 páginas únicas y de la probabilidad de transición entre

³ <http://collectd.org/>

ellas. Existen tres tipos de escenarios posibles: shopping, browsing, y ordering. Para ilustrar el proceso de validación de nuestro benchmark, hemos seleccionado el escenario browsing que se compone de una actividad de navegación muy significativa frente a la escasa actividad asociada a la compra de libros (ordering). Basado en este escenario, el sitio web de validación se reduce a las páginas del proceso de búsqueda (i.e., Home, Search request, Search result y Product detail page) y las transiciones entre ellas.

La Figura 7 detalla el CBMG simplificado para el sitio web de validación, mostrando las diferentes páginas del proceso de búsqueda (Home, Search request, Search result y Product detail page) entre las que pueden transitar los usuarios, y las transiciones permitidas por los arcos del grafo. Los números sobre los arcos indican la probabilidad de transición entre las dos páginas conectadas. Así por ejemplo, la probabilidad de ir a la Product detail page desde la página Search results es del 0.6195. El significado de esta probabilidad hace referencia a que después de una búsqueda, sin tener en cuenta si la misma ha devuelto resultados o no, la Product detail page es visitada en el 61.95 % de los casos. El libro a cuyo detalle se accede será un resultado de la búsqueda o uno de los libros miembros del banner de novedades incluido en la mayoría de páginas de la librería. Estas probabilidades han sido inferidas a partir de los umbrales definidos por la especificación de TPC-W para el escenario browsing. Destacamos que hemos modelado el mismo tipo de carga usando únicamente el concepto de navegación de Dweb y desactivando el resto de parámetros del modelo que nos permiten caracterizar el dinamismo.

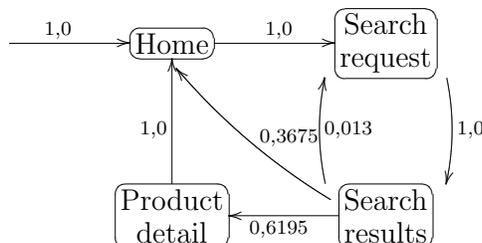


Figura 7. Caracterización de la carga de trabajo en la validación

En las pruebas de validación, hemos contrastado estas dos aproximaciones de caracterización de carga. Únicamente se consideran 50 agentes en la generación de carga debido a que la implementación Java de TPC-W adoptada presenta algunas limitaciones en el proceso de generación, que le impiden generar más carga de forma efectiva aunque consideremos más de 50 agentes. Las medidas han sido realizadas durante 50 ejecuciones con el fin de obtener resultados apropiados con un nivel de confianza del 99 %.

A continuación se presentan los resultados más significativos, de entre todas las métricas medidas en los experimentos, para las cargas definidas con CBMG y Dweb. Ambas cargas generan un número similar de peticiones a

página, como muestra la Figura 8(a). Concretamente, la carga modelada por Dweb produce un 1% menos de peticiones que la carga definida por CBMG, pero esta diferencia no influye en el tiempo de respuesta, que es virtualmente el mismo en ambos casos, tal y como muestra la Figura 8(b).

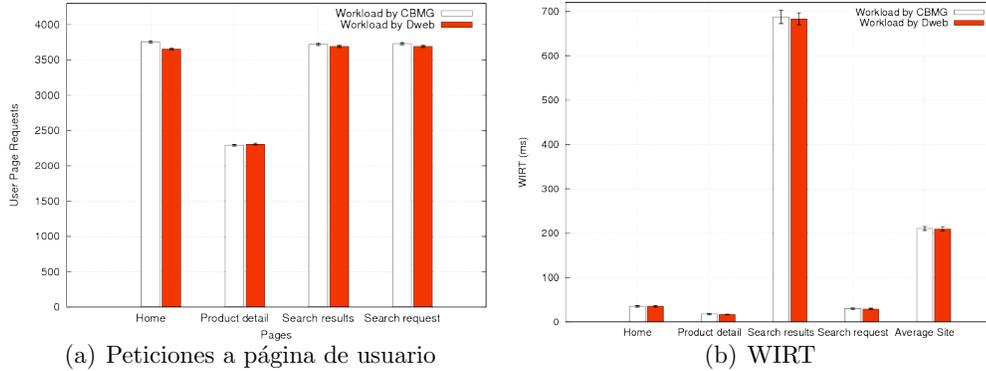


Figura 8. Métricas desde el punto de vista del cliente

Por otro lado, el servidor se caracteriza por un pobre nivel de estrés, representado por una baja utilización de los principales recursos hardware. La Figura 9(a) denota un bajo nivel de utilización de la CPU y la memoria en ambos casos. El tráfico de red entrante y saliente no provoca más de un 3% de utilización de la red de interconexión para ambas cargas, como ilustra la Figura 9(b). Por otra parte, la utilización del disco es demasiado pequeña (inferior al 0.5%) para ser representada gráficamente en ambos casos.

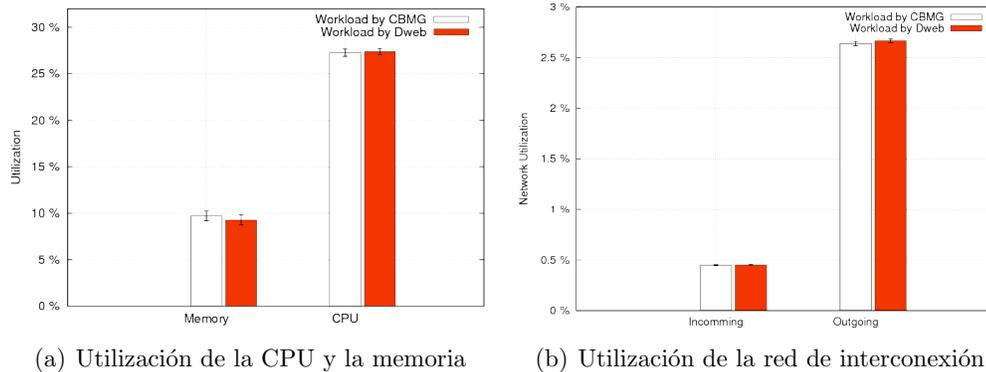


Figura 9. Métricas desde el punto de vista del servidor

Los resultados de las pruebas de validación nos permiten demostrar que Dweb y GUERNICA pueden ser empleados en los estudios de evaluación de prestaciones como alternativa a los modelos de carga tradicionales.

5. Conclusiones

La evolución de la World Wide Web, desde los primitivos servicios estáticos de la primera generación hasta las complejas y personalizadas aplicaciones web actuales, es motivo de otra evolución, la de las pautas de comportamiento de sus usuarios. Usuarios que han dejado de comportarse como meros consumidores de información y han pasado a participar activamente en la creación y difusión de contenidos web dinámicos. Consecuentemente, los usuarios actuales se caracterizan por un comportamiento más dinámico que debe ser contemplado en los modelos de caracterización de carga y en las herramientas que los utilizan para la evaluación de prestaciones web.

En trabajos previos [17] introducimos un nuevo modelo, denominado Dweb, para caracterizar carga web dinámica de una forma más precisa, e implementamos el generador de carga GUERNICA en base a dicho modelo.

El presente trabajo introduce un nuevo entorno de prueba capaz de incorporar generación de carga dinámica en la evaluación de prestaciones de sistemas de comercio electrónico basados en web. Con tal fin, se ha partido del reconocido benchmark de comercio electrónico TPC-W y se ha integrado la generación de carga dinámica proporcionada por Dweb y GUERNICA. El nuevo entorno ha sido validado contra el propio TPC-W.

Como trabajo futuro pensamos demostrar que nuestro modelo de carga es una alternativa más valiosa porque es capaz de reproducir el comportamiento dinámico del usuario en la caracterización de la carga web. Con tal fin, debemos cuantificar el efecto de usar carga dinámica en los estudios de evaluación de prestaciones web.

Referencias

- [1] TPC BENCHMARK(TM) W Specification. Version 1.8. Technical report, Transaction Processing Performance Council, February 2002.
- [2] G. Abdulla. *Analysis and Modeling of World Wide Web Traffic*. PhD thesis, Faculty of the Virginia Polytechnic Institute and State University, Blacksburg, Virginia, May 1998.
- [3] C. Amza, A. Chanda, and A. Cox. Specification and implementation of dynamic web site benchmarks. *IEEE International Workshop on Workload Characterization*, pages 3–13, 2002.
- [4] P. Barford, A. Bestavros, A. Bradley, and M. Crovella. Changes in Web client access patterns: Characteristics and caching implications. *World Wide Web*, 2:15–28, 1999.
- [5] P. Barford and M. Crovella. Generating representative web workloads for network and server performance evaluation. In *ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, pages 151–160. Performance Evaluation Review, July 1998.
- [6] F. Benevenuto, T. R. de Magalhães, M. Cha, and V. A. F. Almeida. Characterizing user behavior in online social networks. In *9th ACM SIGCOMM conference on Internet measurement conference*, pages 49–62, 2009.
- [7] H. W. Cain, R. Rajwar, M. Marden, and M. H. Lipasti. An architectural evaluation of Java TPC-W. In *7th International Symposium on High Performance Computer Architecture*, page 229, 2001.

- [8] G. Cormode and B. Krishnamurthy. Key differences between Web 1.0 and Web 2.0. *First Monday Journal*, 13(6), January 2008.
- [9] R. C. Dodge JR, D. A. Menascé, and D. Barbará. Testing e-commerce site scalability with TPC-W. In *Computer Measurement Group Conference*, pages 457–466, 2001.
- [10] F. Duarte, B. Mattos, J. Almeida, V. A. F. Almeida, M. Curiel, and A. Bestavros. Hierarchical characterization and generation of blogosphere workloads. Technical report, Computer Science, Boston University, 2008.
- [11] S. Floyd and V. Paxson. Difficulties in simulating the Internet. *IEEE/ACM Transactions on Networking*, 9(4):392–403, January 2001.
- [12] D. García and J. García. TPC-W e-commerce benchmark evaluation. *Computer*, 36(2):42–48, February 2003.
- [13] S. Goel, A. Broder, E. Gabrilovich, and B. Pang. Anatomy of the long tail: ordinary people with extraordinary tastes. In *3rd ACM international conference on Web search and data mining*, pages 201–210, 2010.
- [14] D. A. Menascé and V. A. F. Almeida. *Scaling for E-Business: Technologies, Models, Performance, and Capacity Planning*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2000.
- [15] T. O’Reilly. What is Web 2.0. Design Patterns and Business Models for the Next Generation of Software. *O’Reilly Online Publishing*, 2005.
- [16] R. Peña-Ortiz, J. Sahuquillo, J. A. Gil Salinas, and A. Pont. WEB WORKLOAD GENERATORS: A survey focusing on user dynamism representation. In *International Conference on Web Information Systems and Technologies*, May 2011.
- [17] R. Peña-Ortiz, J. Sahuquillo, A. Pont, and J. A. Gil Salinas. Dweb model: representing Web 2.0 dynamism. *Computer Communications Journal*, 32(6):1118–1128, April 2009.
- [18] P. Rodriguez. Web Infrastructure for the 21st Century. In *18th International World Wide Web Conference*, 2009.
- [19] M. Shams, D. Krishnamurthy Ph.D, and B. Far. A model-based approach for testing the performance of web applications. In *International Workshop on Software Quality Assurance*, pages 54–61. ACM, November 2006.
- [20] H. Weinreich, H. Obendorf, E. Herder, and M. Mayer. Off the beaten tracks: exploring three aspects of web navigation. In *15th international conference on World Wide Web, WWW 2006*, pages 133–142, 2006.