

Plataforma de A/B Testing en Redes de Entrega de Contenidos

Alumno: Guillermo Pereyra Irujo

Directora: Dra. Mariana del Fresno

Codirector: Ing. Pablo Daniel Del Grosso

Introducción

La técnica de A/B Testing se utiliza habitualmente en el desarrollo web para contrastar hipótesis de impacto de cambios del producto en variables de negocio. Consiste en la experimentación aleatorizada y controlada con dos variantes: la variante A o *control* es el producto original, la variante B o *tratamiento* es el producto modificado. Se plantea la hipótesis de que una variación particular del producto tendrá una influencia positiva en una variable de negocio y se la expone a un subconjunto de los usuarios, mientras se mide la variable en el conjunto completo. Cuando los resultados alcanzan un umbral de confianza, llamado significatividad estadística, la hipótesis se da por comprobada o refutada según corresponda y la variación se hace permanente o se abandona [1].

La técnica permite llevar a cabo experimentos simultáneos, aunque, cuando las variaciones se solapan en el espacio del producto, es necesario hacer ajustes en el procedimiento para evitar dar por comprobada una hipótesis con información insuficiente [2,3]. Asimismo, en la medición de las variables de negocio, se debe tener en cuenta la distribución de las variaciones entre los usuarios para diferenciar la correlación entre los experimentos de los resultados marginales [4].

Existen en el mercado varias herramientas para la aplicación de la técnica de A/B testing. Algunas son provistas en forma de librerías o módulos que se incorporan a la aplicación web, en el lado servidor de la comunicación web. Otras son provistas en forma de scripts que se ejecutan en el navegador web del usuario, en el lado cliente de la comunicación web. En ambos casos, para cada usuario se elige aleatoriamente una variante y se mide su comportamiento respecto de la variable de negocio.

OLX (<http://www.olx.com.ar>) es una plataforma web de anuncios clasificados, donde los usuarios pueden publicar sus anuncios y contactar a otros anunciantes. Cuenta con un sitio

web, un sitio móvil y varias aplicaciones para dispositivos móviles. Está presente en más de cien países y es visitada varios millones de veces por día. Para definir la evolución de su producto, aplica la técnica de A/B Testing.

Para abastecer su tráfico, OLX utiliza entre sus servidores y los clientes una red de entrega de contenidos: un gran sistema distribuido de decenas de miles de servidores globalmente distribuidos que corren sofisticados algoritmos para posibilitar la entrega de aplicaciones distribuidas altamente escalables [5]. Cuando un cliente solicita un objeto web, lo solicita a la red, y ésta al servidor de aplicación. El servidor de aplicación genera el objeto y la red lo entrega al cliente, pero guarda una copia. La siguiente vez que el objeto es solicitado, la red entrega su copia, reduciendo la carga en el servidor de aplicación y el tiempo de transferencia al cliente.

En el uso actual, el habitual además en el mercado, la red de entrega de contenidos no modifica el contenido de los objetos que distribuye, ni crea nuevos objetos derivados de ellos: sólo trafica copias. Es un componente pasivo en términos de generación de contenidos. Sin embargo, hay redes que tienen la capacidad de interpretar órdenes embebidas en las respuestas del servidor de aplicación, desde simples declaraciones como la inclusión de un subdocumento asociado, hasta instrucciones con complejidad de Turing, como la selección y la iteración [6].

Finalmente, los navegadores web usualmente dividen la tarea de presentación de una página entre varios hilos de ejecución, entre los cuales se encuentran, por separado, uno que parsea el documento HTML recibido del servidor y construye un modelo de objetos del documento (*document object model*, o DOM), y otro que lee el DOM y dibuja en la pantalla (*rendering*) [7]. No hay primitivas de sincronización con esos hilos en espacio de usuario.

Motivación

Una herramienta de A/B testing que genere las variaciones en el servidor de aplicación impedirá el uso de la red de entrega de contenidos: cada solicitud de un objeto que haga un usuario deberá llegar al servidor de aplicación, donde se elegirá qué versión del objeto será entregada. La suma de trabajo es la misma, pero el tiempo de transferencia al cliente crecerá significativamente y la carga en los servidores de aplicación será intolerable.

Una herramienta de A/B testing que genere las variaciones en el cliente deberá necesariamente esperar a que los elementos a variar estén accesibles en el DOM, y modificarlos antes de que el hilo de *rendering* los dibuje. Dado que no hay una interfaz de sincronización, la espera debe ser activa y periódica, dejando una condición de carrera que no garantiza que el contenido original no sea visto por el usuario. Aunque con menor criticidad, que la información completa del experimento quede expuesta al usuario sin una razón de negocio también es inconveniente.

Luego, la generación de las variaciones en el lado del servidor tiene consecuencias inaceptables, y la generación en el lado del cliente tiene consecuencias indeseadas. Se necesita una implementación que esté del lado del cliente para el servidor, y del lado del servidor para el cliente, y el componente de arquitectura que cumple esa condición es la red de

entrega de contenidos. No se encontraron soluciones libres ni comerciales que cubrieran este requerimiento, por lo que se decidió encarar el desarrollo de una implementación propia. Ya se ha llevado a cabo una prueba de concepto para comprobar la viabilidad del proyecto con resultado positivo y se continuará con la presente propuesta.

Objetivo

La propuesta de trabajo consiste en el desarrollo de una plataforma de A/B testing cuya herramienta de variación de contenido se ejecute en la red de entrega de contenidos. El servidor de aplicación podrá entregar el conjunto completo de variaciones a la red de entrega y ésta distribuirá las variaciones a los usuarios según sus asignaciones aleatorias particulares. Así, la mayoría de las solicitudes podrán ser satisfechas en la red, conservando la delegación de la carga, y cada usuario recibirá una sola variación sin saber que está formando parte de un experimento.

La plataforma se completa con un componente de recolección de datos, que muestree las asignaciones aleatorias de variaciones a usuarios y los valores de las variables de negocio, y un componente de reporte, que sumalice los resultados orientando la presentación a la asistencia en toma de decisiones. Como valor agregado, se incluye la automatización de la distribución de las asignaciones, de manera que la correcta medición de correlaciones y resultados marginales no requiera del operador conocimientos especializados.

Cronograma de actividades

Relevamiento bibliográfico sobre software de redes de entrega de contenidos, arquitectura de navegadores web y técnicas de contraste de hipótesis y testeo multivariado.

Análisis y diseño de los componentes de arquitectura: generador de variaciones, medidor de variables de negocio, recolector de datos, panel de reportes.

Desarrollo de prototipos de los componentes con el objetivo de reconocer todos los escenarios y mitigar riesgos en el desarrollo.

Refinamiento del software para cumplir con el conjunto completo de requerimientos funcionales y no funcionales.

Diseño y realización de experimentos sobre la plataforma de OLX.

Cierre del desarrollo y redacción del documento final.

Bibliografía

[1] Mendenhall, W.; Beaver, R.; Beaver, B. (2012). "Introduction to probability and statistics". Cengage Learning.

[2] Ioannidis, J. P. A. (2005). "Why Most Published Research Findings Are False". PLoS Med 2(8): e124.

[3] Johnson, R. A.; Wichern, D. W. (2002). "Applied multivariate statistical analysis". Upper Saddle River, NJ: Prentice hall, 5(8).

[4] Cox, D. R. (1958). "Planning of experiments". Oxford, England: Wiley.

[5] Nygren, E.; Sitaraman, R. K.; Sun, J. (2010). "The Akamai Network: A Platform for High-Performance Internet Applications". ACM SIGOPS Operating Systems Review 44(3): pp. 2-19.

[6] Tsimelzon, M.; Weihl, B.; Chung, J.; Frantz, D.; Brasso, J.; Newton, C.; Hale, M.; Jacobs, L.; O'Connell, C. (2001). "ESI Language Specification 1.0". W3C. <http://www.w3.org/TR/esi-lang>. Retrieved 2014-02-12.

[7] Grosskurth, A.; Godfrey, M. W. (2005). "A reference architecture for Web browsers". Proceedings of the 21st IEEE International Conference on Software Maintenance, 26-29 Sept. 2005, pp. 661-664.