



Identificación del Trabajo	
Área:	Sistemas de información e informática
Categoría:	Alumno
Regional:	Santa Fe

Herramienta para la evaluación de performance de infraestructuras de e-commerce en nubes computacionales privadas.

Laura Doval Temporelli

Departamento Ing. En Sistemas de Información (Lavaise 610, Santa Fe), Facultad Regional Santa Fe, UTN

E-mail de autores: lauradt@gmail.com

Este trabajo ha sido realizado bajo la dirección de la Dra. Ana Rosa Tymoschuk y la co-dirección del Dr. Carlos Chezzi, en el marco del proyecto PID UTN4855 "Desarrollo de frameworks para la generación y evaluación de infraestructuras tecnológicas con usos de nubes computacionales para comercio electrónico". (2018 – 2019)

Resumen

Los avances de los sistemas de información permiten disponer de infraestructuras tecnológicas para el comercio electrónico. Las tecnologías de computación en la nube son muy populares para el e-commerce. La nube ofrece servicios cuyos costos se calculan de acuerdo al uso de los recursos que ofrece, en general, máquinas virtualizadas. De esta forma no es necesaria una inversión inicial en infraestructura. En este trabajo se presenta una infraestructura desarrollada en una nube privada, con el fin de estudiar la performance tecnológica y su relación con los beneficios económicos. La idea es comparar costos, alcances, ventajas y desventajas de e-commerce usando nubes públicas y privadas. Se implementa un sitio de e-commerce en una plataforma de máquinas virtuales y evalúan escenarios con la herramienta J-Meter para obtener métricas necesarias para asociar costos, para tomar decisiones sobre el tipo infraestructura a adoptar para el modelo de negocios electrónicos.

Palabras Claves: e-commerce; nube privada; performance; métricas.

1 Introducción

Las nubes computacionales se consideran una estrategia de negocio para el procesamiento de transacciones comerciales mediante internet, proveen servicios a través de recursos virtualizados, estos servicios se dividen en tres categorías:

(i) Software como servicio. (SaaS: Software as a Service), se contrata la utilización de aplicaciones que se ejecutan en la infraestructura de la nube computacional.

(ii) Plataforma como servicio. (PaaS: Platform as a Service), en la cual se proveen los recursos informáticos; como bibliotecas, servicios, herramientas, etc; para desarrollar aplicaciones en la infraestructura de la nube computacional.

(iii) Infraestructura como servicio. (IaaS: Infrastructure as a Service), proporciona la capacidad de procesamiento, almacenamiento y redes para implementar y ejecutar un software determinado como sistema operativo y aplicaciones.[1]

Los usuarios pueden contratar estos servicios en lugar de comprar equipamiento o desarrollar sus propias aplicaciones. Con la integración del comercio electrónico y la nube computacional se logra una estructura flexible para la implementación de aplicaciones y con un costo asociado a lo que se consume del servicio de la nube. De esta manera el usuario puede adaptar la tecnología que necesita para realizar transacciones comerciales buscando siempre el beneficio económico. Hay una relación entre performance tecnológica con beneficios económicos, [2] por lo que es de interés conocer los costos asociados con la infraestructura de la tecnología de la información (IT) y su desempeño durante las transacciones comerciales para poder estimar ingresos económicos y beneficios comerciales.

Es importante realizar simulaciones de cargas de trabajo en el sitio de comercio electrónico, para representar el comportamiento de los clientes. Los resultados sirven como indicadores tanto tecnológicos como económicos. Además estas experimentaciones permiten predecir comportamientos en distintos escenarios como por ejemplo el incremento la cantidad de clientes, de productos, logística de entrega, etc.[3]

En un trabajo de investigación previo se implementó una aplicación en la nube de Amazon en la capa gratuita de prueba para obtener indicadores.[4]

Debido a la complejidad en las implementaciones para simulaciones y la obtención de parámetros para modelos de simulación de sitios de negocios electrónicos para predecir comportamientos tecnológicos y económicos, así como los aspectos de seguridad de información y riesgos de fallas de las nubes, se orientó la investigación analizando otras alternativas como la de sistemas distribuidos virtuales de infraestructuras propietarias, para obtener los mejores desempeños y rentabilidad. [5]

En este trabajo de investigación se describe una plataforma de virtualización como servicio de infraestructura para comercio electrónico para poder analizar su rendimiento tecnológico como alternativa a los recursos ofrecidos por las nubes computacionales públicas. Esta plataforma se implementa mediante un esquema de máquinas virtuales, en el que se instala el sitio de negocios mediante la aplicación Prestashop.

El desempeño del sitio de comercio electrónico se mide mediante el generador de carga y monitor JMeter “jmeter.apache.org”. Las métricas obtenidas se analizan y comparan con las nubes públicas para obtener conclusiones que orienten a estimar beneficios comerciales.

El objetivo es presentar una plataforma tecnológica de generación de infraestructuras virtuales para la implementación de una aplicación de comercio electrónico en un contexto de nube privada y proponer una estrategia para evaluar su desempeño y poder en otro trabajo calcular sus costos, para compararlos con las nubes públicas y recursos propietarios.

2 Metodología

2.1 Implementación de un sitio de comercio electrónico de prueba

Para implementar una Tienda Virtual de Comercio Electrónico y realizar pruebas de performance y comparar costos, se eligió la aplicación Prestashop [6], de licencia libre, (<https://opensource.org/licenses/OSL-3.0>), la cual está escrita en el lenguaje de programación PHP y utiliza el motor de base de datos MySQL. Requiere de un servidor web Apache y del lenguaje de programación PHP instalado en el entorno donde va a ser desplegada.

La aplicación ofrece las siguientes funcionalidades: creación de productos y paquetes de productos, navegación de productos, control de stock, gestión de proveedores, configuración de la tienda, gestión de pagos, gestión de envíos, buscador de productos, geolocalización, configuración de normativas legales de cada país, creación de ofertas especiales y cupones de descuento, publicidad de la tienda, control de los pedidos e informes, entre otros.

En este trabajo se analiza la estructura y el funcionamiento de la aplicación y se toman en cuenta las siguientes transacciones para poder realizar las pruebas de funcionamiento sobre la misma:

2.1.1 Caso 1: El usuario entra a la página principal de la tienda

El usuario ingresa a la página principal del sitio y no realiza ninguna otra interacción con la aplicación.

2.1.2 Caso 2: El usuario utiliza el buscador e ingresa al detalle de un producto en particular

El usuario utiliza el buscador del sitio, ingresa qué tipo de producto busca y una vez que se presiona la lupa del buscador o “enter”, el sitio muestra los resultados de la búsqueda con las opciones de productos posibles según lo ingresado; de esta manera, el usuario puede seleccionar un producto, permitiéndole visualizar la imagen del producto, sus opciones de talle, color y tiene la opción de ampliar el detalle para obtener más información.

2.1.3. Caso 3: El usuario busca el producto y procede con la compra

El usuario realiza las dos transacciones antes mencionadas, ingreso al sitio y búsqueda de producto, pero además compra el mismo, pudiendo seleccionar en la pantalla donde se muestra el producto elegido, talla, color y cantidad. Clickeando sobre el botón “add to cart”, el sitio muestra una ventana desplegable donde se visualiza el detalle del producto elegido, el monto a pagar y datos de lo seleccionado. Luego, el usuario puede continuar con la compra.

2.2 Infraestructura usando máquinas virtuales

Para implementar la infraestructura donde se ejecuta la aplicación Prestashop se utilizó un enfoque denominado “Infrastructure as code”. Este define la arquitectura de servidores utilizando código de programación Ruby (<https://www.ruby-lang.org/>) y la misma es creada cuando se ejecuta el mismo. Una vez desplegada, ésta puede ser accedida por los clientes.

En este enfoque la definición de la infraestructura se trata como software, por lo que pueden aplicarse técnicas propias del desarrollo de software como Versionado, Integración Continua, Revisión de Código y Testing Automatizado.

De esta forma se pueden entregar productos de software de una manera sostenida en el tiempo y confiable, ya que se tiene conocimiento pleno de la infraestructura y no se incurre en posibles errores de modificaciones manuales en cada servidor. Para realizar cambios de arquitectura, se modifica el código, se ejecuta la definición y la nueva infraestructura es desplegada para los clientes. Esto facilita la detección de posibles problemas de performance o de lógica de negocios y las correcciones necesarias.

Esta metodología permite replicar y automatizar las infraestructuras de IT en diferentes proveedores que pueden ser servidores físicos o la nube.

En las pruebas de implementación de infraestructura con máquinas virtuales se utilizó un servidor HP PROLIANT DL380 que cuenta con 16 CPUs de 3.5 GHZ y 32 Gb de RAM DDR4. Esta configuración de hardware sirve para realizar pruebas a escalas significativas respecto a las de nubes públicas. El sistema Operativo base es Debian 8. El servidor está instalado en la ciudad de Concordia, provincia de Entre Ríos y los requerimientos provinieron de usuarios distribuidos en Santa Fe Capital y en Concordia.

Para implementar la infraestructura se utilizó la tecnología Vagrant[7], que es una herramienta de software para el manejo y gestión de máquinas virtuales.

Las máquinas virtuales necesitan, una vez creadas, ejecutar scripts de instalación de los programas y librerías correspondientes al perfil de la máquina a generar para diferenciarlas en sus funciones. Para esta etapa se eligió la tecnología Ansible [8], que es una herramienta de software de automatización de tareas para administración y configuración de múltiples máquinas físicas o virtuales.

La arquitectura planteada donde se ejecuta la aplicación de e-commerce consiste en 3 tipos de servidores:

- **Balanceadores de carga:** los cuales reciben las peticiones HTTP y las redirecciona hacia los servidores Web siguiendo un algoritmo Round Robin.
- **Servidores Web:** que son los responsables de recibir peticiones de los balanceadores de carga y generar respuestas HTTP hacia el usuario. Ejecutan el código PHP de la aplicación Prestashop.

- **Servidores de base de datos:** donde se ejecutan los motores de base de datos utilizados desde los Servidores Web.

Esta configuración de las máquinas virtuales permite realizar modificaciones, generando distintos escenarios para representar situaciones en un sitio de comercio electrónico y recopilar datos con una herramienta adecuada para luego obtener conclusiones sobre la arquitectura de servidores óptima para cada escenario.

2.3 Generador de carga de trabajo o de stress sobre la aplicación de e-commerce

Para las experimentaciones en el sitio de e-commerce se utilizó la herramienta JMeter de Apache, que permite realizar pruebas funcionales y medir la performance de recursos estáticos y dinámicos. Mediante las pruebas de simulación se pudo observar el desempeño del sistema y su comportamiento para así poder hacer los ajustes necesarios para mejorar la eficiencia.

JMeter es una herramienta para efectuar pruebas de estrés y de rendimiento y encontrar los cuellos de botella en aplicaciones web. También permite generar diversos tipos de reportes en los que se incluyen gráficos y tablas que muestran las métricas de desempeño.

Mediante la herramienta JMeter se implementaron los tres casos descritos en la sección 2, denominados clases (clase 1, clase 2, clase 3), variando el número de hilos (Usuarios) para simular diversas situaciones de ingresos al sitio de e-commerce, para una instancia o máquina virtual.

Una vez generada la infraestructura virtual, se procedió a experimentar los casos de prueba o de stress con el objetivo de obtener resultados tendientes a identificar la mejor configuración para las necesidades computacionales deseadas, en términos de performance de recursos tecnológicos (utilizaciones de los recursos, tiempos de respuesta, velocidades de procesamiento, ocurrencia de eventos de fallas) y de negocios (número de sesiones de clientes exitosas, cantidad de ventas, beneficios económicos, entre otros).

. Cada clase tiene un “grupo de hilos”, cuya cantidad representa a los clientes que realizan las mismas transacciones en un tiempo de generación determinado.

Cada “grupo de hilos” se comporta diferente de los otros dos, pero dentro del grupo, los hilos ejecutan las mismas transacciones. En el grupo de hilos se configura el número de hilos (cantidad de usuarios), el período de subida (cuantos segundos se tardará en alcanzar el número máximo de usuarios) y el contador del bucle (número de veces que cada hilo ejecutará la prueba). Se plantearon escenarios considerando un tiempo de generación de transacciones o tiempo entre arribos (ramp-up) de 10 segundos, según se puede ver en la Tabla 1:

Tabla I. Configuración de las pruebas.

Escenario	Clase de trabajo	Cantidad de hilos
0	1	200
1	2	80
	3	30

	1	300
2	2	150
	3	50

Se desarrollaron scripts para que cada hilo genere distintos datos a ser usados en la navegación. Así, a pesar de que el comportamiento sea el mismo para cada hilo dentro de un mismo grupo, los datos tales como el producto a buscar, el nombre de usuario y/o el email, entre otros, serán distintos (Figura 1).

```
Script (variables: ctx vars props prev data log)
Script
1 String nombre = vars.get("nombre");
2 String apellido = vars.get("apellido");
3 String nombreApellido = nombre + apellido;
4 int longitud = nombreApellido.length();
5 String email = nombreApellido.charAt(0).toString();
6
7 for(int i = 1; i < longitud; i++){
8     if(Math.random() > 0.7){
9         email += nombreApellido.charAt(i).toString();
10    }
11    else {
12        email += ((int)Math.floor(Math.random() * (211 - 1)) + 1).toString();
13    }
14 }
15
16 email += ((int)Math.floor(Math.random() * (2001 - 1)) + 1).toString();
17 email += "@gmail.com";
18
19 vars.put("email", email);
```

Fig. 1: Scripts de navegación.

Las aplicaciones de registro de datos o muestreadores contienen la url a la que se quiere enviar una petición con sus respectivos parámetros, el puerto y el protocolo. Para simular todo el flujo de navegación de un usuario en una determinada clase se agregó un muestreador por cada tipo consulta http que se requiere. El análisis de resultado se realiza a partir del "Reporte Resumen" de la herramienta JMeter, en el cual se presentan las siguientes métricas:

- **Muestras:** número de solicitudes enviadas por grupo de hilos.
- **Media:** es el tiempo de respuesta promedio, contabilizado como el tiempo transcurrido desde que una solicitud es enviada al servidor hasta el momento en que la misma retorna a quien la solicitó.
- **Min:** es el tiempo mínimo de respuesta (milisegundos).
- **Máx:** es el tiempo máximo de respuesta (milisegundos).
- **Desv. Estándar:** Desviación estándar de la muestra por hilo (milisegundos).
- **% Error:** porcentaje de pruebas que fallaron.
- **Rendimiento:** es el número de solicitudes por unidad de tiempo que se envían al servidor durante la prueba.
- **kb/sec:** es el rendimiento medido en kilobytes por segundo.
- **Media de Bytes:** tamaño promedio de la respuesta en bytes.

3 Resultados

En la Tabla 2 se presenta un resumen de métricas del "Reporte Resumen" resultante de las experimentaciones realizadas en JMeter. Para el caso del escenario 1 se computa un total de 650 muestras, mientras que para el dos se observan 1045 muestras. Se observa que para un tiempo

de generación de transacciones o ramp-up de 10 segundos, en los dos escenarios propuestos se obtienen métricas de tiempo de respuesta adecuadas y con un 0 % de error. Por lo tanto se puede concluir que el servidor no se llega a saturar con las cargas experimentadas para una sola instancia. La siguiente etapa será considerar más instancias y mayor carga de trabajo para observar cómo se comporta el sistema con el incremento de la capacidad.

Tabla II. Resultados de las pruebas.

Escenario	Operación	Rendimiento	Tiempo de Respuesta	Error
1	Entrar al Index (clase 1)	2,4 req/seg	50,6 seg	0 %
	Entrar al detalle (clase 2)	39.3 req/min	5,25 seg	0 %
	Confirmar compra (clase 3)	1,3 req/seg	2,22 seg	0 %
2	Entrar al Index (clase 1)	3,1 req/seg	60,20 seg	0 %
	Entrar al detalle (clase 2)	43,0 req/min	12,26 seg	0,5 %
	Confirmar compra (clase 3)	1,9 req/seg	2.79 seg	0 %

4 Discusión

En este trabajo se presentó la implementación de una infraestructura de virtualización para cargas de trabajo de un sitio de comercio electrónico. Se utilizaron herramientas de software libre para la virtualización y para el sitio web de compras por internet. Se logró configurar la carga de trabajo consistente en tres clases de trabajo con diferentes cantidad de hilos mediante la herramienta J-Meter para la simulación, que además posibilita la evaluación de acuerdo a cargas y la obtención de métricas que aportan los fundamentos para el análisis de desempeño.

Como resultados para una instancia, como primera etapa de pruebas se obtuvieron rendimientos y tiempos de respuesta, con incrementos más significativos en esta última métrica. No obstante en ambos escenarios se observaron tasas de errores casi nulos y el procesamiento de todas las clases que se generaron en los 10 segundos de ramp-up.

Como trabajo futuro se prevé la extensión de la plataforma a más de una máquina virtual y con un mayor número de escenarios de experimentación, con distinta cantidad de usuarios por caso o clase.

Respecto a la herramienta JMeter se incorporarán nuevas métricas de salidas como las ofrecidas por el “Informe Agregado” y el de “Gráficos Agregados”, así como también diferentes cargas para provocar la saturación del sistema y poder analizar el comportamiento del sistema ante el estrés ante más de una máquina virtual.

Los resultados de evaluación de performance para distintos escenarios y cargas de trabajo o clases y número de máquinas virtuales se relacionarán con costos para identificar la configuración

más conveniente, tanto desde el desempeño (tiempos de respuestas, rendimiento, etc) como desde los beneficios económicos.

5 Conclusiones

Mediante la herramienta JMeter se lograron realizar simulaciones que permitieron analizar la performance de un sistema de comercio electrónico mediante métricas a partir de diferentes cargas de usuarios y comportamientos que permitirán anticiparse a las distintas demandas del mercado.

Reconocimientos

La realización de este trabajo se hizo en el marco del PID UTN ENUTIFE0004855TC “Desarrollo de frameworks para la generación y evaluación de infraestructuras tecnológicas con usos de nubes computacionales para comercio electrónico”, el cual recibió financiamiento de la UTN y de la SPU (Ministerio de Educación). Se agradece al equipo de trabajo del proyecto, Federico Baroni, Ricardo Lerman, Ana Rosa Tymoschuk, Carlos María Chezzi y Martín Domínguez quienes han colaborado con las experiencias y la configuración de la infraestructura

Referencias

1. Mell, P., Grance, T., “The NIST Definition of Cloud Computing.” National Institute of Standards and Technology (NIST). Special Publication (2011) 800-145.
2. Chezzi, C. M. “Modelado y Simulación de Desempeño de Procesos de Comercio Electrónico”. Universidad Tecnológica Nacional. Facultad Regional Santa Fe. Santa Fe, Argentina. Tesis Doctorado en Ingeniería Mención Ingeniería de Sistemas de Información
3. Lerman, R., Viviani, B. “Modelo de simulación para evaluación de desempeño de la plataforma computacional de comercio electrónico”. Jornada de Jóvenes Investigadores JIT 2015, UTN Facultad Regional Venado Tuerto, Venado Tuerto, Santa Fe (2015).
4. Basualdo, R., Massuh, E., Valentinis, F. “Estudio del comportamiento de un modelo de negocios B2C en la nube computacional”. Jornada de Jóvenes Investigadores JIT 2016, UTN Facultad Regional Santa Fe, Santa Fe (2016).
5. Baroni, F., Basualdo, R., Massuh, E., Domínguez, M., Tymoschuk, A., Chezzi, C. “Desarrollo de una plataforma para la evaluación de infraestructuras Cloud Computing aplicadas para e-commerce”. CONAISI 2017, Santa Fe, (2017).
6. Prestashop: <https://devdocs.prestashop.com/1.7/basics/installation/system-requirements/>. Último acceso mayo de 2019.
7. Vagrant: <https://www.vagrantup.com>. Último acceso mayo de 2019.
8. Red Hat Ansible: <https://www.ansible.com/overview/it-automation>. Último acceso mayo de 2019.