

Proyecto - Entrega No. 1

Análisis de Capacidad – Plan de Pruebas de Carga

Objetivos

Diseñar y documentar un plan de pruebas de carga y estrés para evaluar la capacidad máxima de los componentes del sistema, incluyendo la infraestructura subyacente.

- Seleccionar e instalar herramientas de prueba de carga y estrés adecuadas al contexto del proyecto.
- Definir escenarios clave de prueba que simulen rutas críticas del usuario.
- Identificar y documentar el entorno de pruebas, incluyendo hardware, software y red.
- Establecer criterios de aceptación para throughput, tiempos de respuesta y utilización de recursos.
- Ejecutar pruebas preliminares con herramientas que generen carga a aplicaciones web para validar el entorno.
- Documentar los hallazgos iniciales y preparar la infraestructura para pruebas futuras.
- Para esta primera entrega, el resultado esperado es el documento del plan de pruebas de carga y análisis de capacidad.
- Seleccionar una herramienta para ejecutar las pruebas de estrés que permitan dimensionar la capacidad de una aplicación y su infraestructura de soporte con base al plan de pruebas, para las futuras entregas.

Lecturas previas

- Tutorial Análisis de Capacidad
- Tutorial de Pruebas de Estrés

Lugar y formato de entrega

Para cada una de las entregas del proyecto se deben proporcionar los siguientes recursos:

- Crear la carpeta **pruebas-carga** dentro del repositorio del proyecto.
- En la nueva carpeta, crear un documento en formato Markdown llamado **plan_de_pruebas.md**.
- Especificar la herramienta de pruebas de carga que se utilizará para las entregas 2 a 5.
- Describir el diseño del entorno de pruebas (por ejemplo, la infraestructura requerida para ejecutar las pruebas de carga).
- Especificar los criterios de aceptación.
- Topología de la prueba de carga (cliente, servidor, red).
- Diagrama de flujo de un escenario de prueba.

Análisis de capacidad

En esta sección, presentamos Apache JMeter como una herramienta para llevar a cabo sus pruebas de carga. No obstante, puede optar por otras herramientas según sus preferencias y experiencia (Apache Benchmark, Gatling, Locust, etc). Utilizamos Apache JMeter como referencia para ilustrar algunos conceptos y la recomendamos por su simplicidad.

Apache JMeter es una aplicación de código abierto, desarrollado en Java. JMeter es una herramienta diseñada para realizar pruebas de estrés de aplicaciones Web HTTP. Es importante resaltar que **JMeter no es un navegador web, es una herramienta que simula las peticiones a nivel del protocolo HTTP, por esta razón no compila el código embebido**. JMeter facilita simular la carga de cientos de usuarios de una aplicación web.

Apache Bench (ab) es una herramienta de prueba de carga y evaluación comparativa para servicios basados en web. Se puede ejecutar desde la línea de comandos y es muy simple de usar. Se puede obtener una salida de prueba de carga rápida en segundos. Como no necesita demasiada familiaridad con los conceptos de prueba de carga y rendimiento, es adecuado para usuarios principiantes e intermedios. Para usar esta herramienta, no se requiere una configuración compleja. No tiene todas las características de herramientas más populares como JMeter o Grinder, pero es una buena herramienta para empezar.

Las pruebas de estrés evalúan: capacidad de procesamiento, tiempos de respuesta, y utilización. Estas tres métricas son indispensables para diseñar pruebas de estrés efectivas.

- **Capacidad de Procesamiento (Throughput):** la cantidad de trabajo realizado en un periodo definido de tiempo (Ejemplo: transacciones por minuto). En JMeter, el listener Summary Report reporta el throughput promedio para la totalidad de las transacciones y para cada una individualmente.
- **Tiempo de Respuesta (Response Time):** tiempo que transcurre desde que el usuario envía una transacción hasta que recibe la respuesta completa. En JMeter, el listener Summary Report reporta el Response Time promedio para la totalidad de las transacciones y para cada una individualmente.
- **Utilización (Utilization):** porcentaje que representa qué tan ocupado está un recurso (CPU, Mem, I/O, etc) en un momento específico. JMeter no reporta esta métrica; es necesario instalar el plugin Performance Monitoring pero se recomienda utilizar un Application Performance Management (APM) independiente.

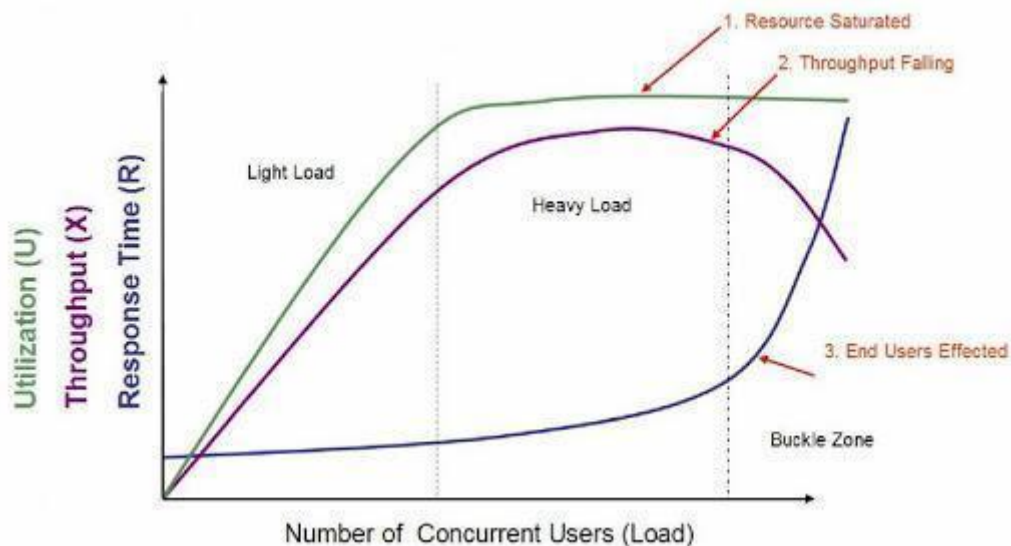


Ilustración 2. Ejemplo, comportamiento de las métricas en un escenario de estrés.

Recuerde, las pruebas de estrés deberán responder a interrogantes como: ¿la aplicación responde rápidamente? ¿cuál es volumen de usuarios puede manejar la aplicación con la infraestructura que actualmente la soporta?

En la ilustración 2, puede analizar el comportamiento de las tres métricas mencionadas en función de la carga artificial creada (simulación de usuarios). Los incrementos en la carga impactan la capacidad de procesamiento, el tiempo de respuesta y utilización. Inicialmente, la relación tiende a ser lineal con baja carga; a medida que la carga aumenta hacen que el comportamiento de las métricas sea no-lineal.

Este ejemplo ilustra la saturación de una CPU cuando el número de procesos supera la capacidad de la CPU. El tiempo de respuesta aumentará a razón del número de procesos en cola, y como consecuencia, genera que la capacidad de procesamiento disminuya y se afecte la experiencia de un usuario.

Para todas las entregas del proyecto usted deberá realizar pruebas de estrés, con el objetivo de comprender cómo la aplicación web y su infraestructura responden a varios niveles de carga de usuarios. Con estas pruebas se desea medir la capacidad de procesamiento, el tiempo de respuesta y utilización, mientras se simula la concurrencia de usuarios en la aplicación y cargas de trabajo artificiales.

Para esta primera entrega no es necesario que usted este familiarizado con JMeter (en semanas posteriores trabajaremos sobre la herramienta). Para esta entrega puede utilizar Apache Bench (ab) para el desarrollo de las pruebas de carga. Sin embargo, es una decisión que cada grupo puede tomar libremente.

Dado esto es necesario diseñar una carga efectiva (estrés) e interpretar los resultados correctamente, para esto responda las siguientes preguntas orientadoras:

1. ¿Cuál es su entorno de prueba? Identifique las características y limitaciones de la infraestructura donde se despliegue su aplicación en producción, así como las características de las herramientas que soportan su aplicación. Es necesario identificar estos aspectos para el equipo de prueba. El entorno físico incluye configuraciones de hardware, software y red. Tener un conocimiento profundo de todo el entorno de pruebas desde el principio permite un diseño y una planificación de pruebas más eficientes y le ayuda a identificar los desafíos de las pruebas al principio del proyecto.
2. ¿Cuáles son los criterios de aceptación? Identifique los objetivos y limitaciones de tiempo de respuesta, rendimiento y utilización de recursos. El tiempo de respuesta es una preocupación del usuario, el rendimiento es una preocupación comercial y la utilización de recursos es una preocupación del sistema. Además, identifique

los criterios de éxito del proyecto que pueden no ser capturados por esos objetivos y limitaciones; por ejemplo, utilizando pruebas de rendimiento para evaluar qué combinación de ajustes de configuración dará como resultado las características de rendimiento más deseables. Valide la información que hemos definido para los escenarios de prueba.

3. ¿Cuáles son los escenarios de prueba? Identificar escenarios clave, determinar la variabilidad entre servicios representativos y cómo simular esa variabilidad, definir datos de prueba y establecer qué métricas se deben recopilar. Consolide esta información en uno o más modelos de uso del sistema para implementar, ejecutar y analizar.

Ejemplos de elementos a considerar en la estrategia de pruebas:

- 3 etapas: Prueba de humo, prueba de carga progresiva, prueba de estrés.
 - Cargas incrementales: 10, 50, 100, 500 usuarios concurrentes.
 - Criterios de éxito:
 - Tiempo de respuesta promedio < 500ms (Web)
 - Throughput > 200 transacciones/minuto (Batch)
4. ¿Cuáles son los parámetros de configuración? Prepare el entorno de prueba, las herramientas y los recursos necesarios para ejecutar cada estrategia a medida que las características y los componentes estén disponibles para la prueba. Asegúrese de entender los requerimientos, las limitaciones y las restricciones de Apache Bench (ab) o JMeter y su APM.

Ejemplos:

- Parámetros del entorno (CPU, RAM, Red)
 - Herramientas de monitoreo adicionales (por ejemplo, CloudWatch + Grafana)
 - Métricas objetivo
5. Identifique **dos escenarios clave** para realizar pruebas de carga en las próximas entregas. Se recomienda que estos escenarios estén enfocados en las acciones que los usuarios realizan al utilizar la aplicación, utilizando la idea de 'rutas críticas del usuario' para capturar la experiencia del cliente. Estas rutas consisten en secuencias de tareas esenciales para el servicio, como la búsqueda de un

producto, la adición de productos al carrito de compras y la finalización de una compra. Aunque medir estas tareas puede ser complicado, es fundamental identificar lo que es importante para el usuario antes de garantizar el rendimiento y la confiabilidad de esa parte del servicio. Una vez que se hayan identificado los escenarios centrados en el usuario, se pueden medir y rastrear como Indicadores Clave de Rendimiento (SLIs) y Objetivos de Nivel de Servicio (SLOs). Con base en esta descripción, identifique un escenario relacionado con la capa web y otro relacionado con la capa de procesamiento por lotes.

Ejemplos:

- **Ruta Crítica 1:** Inicio de sesión -> Búsqueda de producto -> Compra.
- **Ruta Crítica 2:** Procesamiento nocturno de lotes -> Carga masiva de datos.

6. Indique la herramienta que utilizará para llevar a cabo las pruebas de carga y describa las capacidades de la infraestructura requerida para su ejecución. La infraestructura se refiere a una instancia de cómputo en Amazon EC2 que se utilizará exclusivamente para realizar las pruebas de carga en las entregas posteriores, actuando como cliente o usuario simulado. Para esta infraestructura, se hará uso de los recursos disponibles en AWS Academy, específicamente el Learner Lab. Como parte de este documento, es necesario dimensionar las capacidades de CPU, memoria y disco de esta instancia de cómputo en función de su plan de pruebas y de la herramienta que seleccione para llevar a cabo dichas pruebas.

Ejemplo de la infraestructura de Pruebas:

- Instancia EC2: m5.large (2 vCPU, 8 GB RAM)
- Justificación:
 - Soporta hasta 500 usuarios concurrentes simulados en Apache JMeter.
 - Baja latencia de red en región us-east-1.
- Monitoreo:
 - CloudWatch Metrics: CPU, Red, EBS IO.
 - JMeter Backend Listener con InfluxDB + Grafana para análisis de métricas.

Consideraciones importantes

1. Agregar una **tabla resumen de los escenarios y resultados**.
2. Incluir **gráficos de resultados preliminares** aunque sean simulados:
 - a. Throughput vs. Usuarios
 - b. Tiempo de Respuesta vs. Carga
 - c. Entre otros más relacionados con sus métricas y criterios de aceptación
3. Adjuntar los **scripts de prueba**.