ENTREGA 5 - ANÁLISIS DE CAPACIDAD

Presentado por:

- Esteban Emmanuel Ortiz Morales ─ 201913613
- Andrés Martin Ochoa Toro ─ 201913554
- Daniel Jiménez ─ 201011658
- Manuel Felipe Porras Tascón ─ 201913911

1. **(20%)** ¿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.

Lo primero que debemos tener encuentra es que, para el despliegue en producción de la aplicación, realizará un despliegue en la nube.

Particularmente, en la industria es un estándar que las instancias de la capa gratias tengan las siguientes características: 1 vCPU y 1 GB de memoria RAM, por lo tanto, debemos siempre tener en cuenta las limitaciones de recursos que nos ofrece este entorno.

De la misma manera, la versión seleccionada del sistema operativo en el cual hacer el despliegue es Ubuntu 22.04 LTS pues los requisitos de librerías y herramientas de la aplicación como celery, redis y gunicorn son especializados para este sistema operativo.

Además , tenemos tres grupos de instancias , dos que son web servers y estos se van a distribuir la carga mediante un balanceador de cargas que escalan de 1 a 2 máquinas más y otro donde actuará y escalará el worker cuando sea necesario y la carga de las peticiones por parte de los usuarios se lo exija escalando de 1 a 3 máquinas más.

Los operadores de la nube, ofrecen ciertas limitaciones de red. Específicamente tienen un ancho de banda limitado y un número máximo de operaciones entrada/salida. Por ello es importante tener en cuenta estas limitaciones y ajustar la configuración de red de la aplicación según se requiera.

Finalmente, para las herramientas de prueba de carga sobre la aplicación, consideramos el uso de Apache JMeter. Esta aplicación es compatible con el sistema operativo de Ubuntu y presenta facilidad para realizar la configuración en entornos de despliegue en la nube y configuraciones de red específicas.

2. **(20%)** ¿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.

Para definir los criterios de aceptación respecto a las peticiones de la aplicación desarrollada. Se pueden considerar los siguientes aspectos:

1. **Subida de archivos:** La aplicación debe permite al usuario subir archivos en diferentes formatos (.txt, .docx, pdf, jpg, etc) y tamaños. De igual manera, la subida de archivo debe ser exitosa y los archivos deben ser almacenados correctamente en la base de datos postgres definida. En cuanto a tiempos de respuesta, para un archivo promedio (10 MB) la aplicación debe permitir subirlo en menos de 4 segundos.
2. **Descarga de archivos**: La aplicación debe permitir al usuario descargar los archivos subidos previamente en cualquier formato que se haya comprimido, o bien el formato original. La descargar de archivos debe ser exitosa y los archivos deben estar disponibles en el formato correcto para poder descargarlos sin errores. El tiempo de respuesta definido para la descargar de un archivo promedio es de 6 segundos.
3. **Compresión de archivos:** La aplicación debe permitir al usuario comprimir los archivos subidos en diferentes formatos como .zip, tar.gz, rar. La compresión del archivo debe ser exitosa, el archivo comprimido debe estar completo y sin errores y disponible para descargar. El tiempo de respuesta definido es de 10 segundos.
4. **Funcionalidad general de la aplicación:** Peticiones como la de login o listar tareas de compresión, son computacionalmente menos exigentes que las peticiones previamente definidas y de igual manera cumplen un rol vital. Se espera que la aplicación funcione de manera eficiente, sin demoras excesivas ni tiempos de espera prolongados al usuario. El tiempo de espera máximo definido es de 3 segundos.

3. **(20%)** ¿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.

Decidimos plantear tres escenarios de prueba que dependen de la cantidad de usuarios virtuales. Por lo tanto, se definió el siguiente esquema de acuerdo a las limitaciones del entorno de producción.

- **Alta demanda de la aplicación:** 100 usuarios concurrentes
- **Demanda Intermedia de la aplicación:** 50 usuarios concurrentes
- **Demanda baja de la aplicación:** 10 usuarios concurrentes

Estos tres escenarios se probarán en un instante específico y de igual manera en condiciones de uso continuo. Las pruebas definidas son las siguientes:

1. **Subida y descarga de archivos:** Se deben probar los tiempos de respuesta y la capacidad de la aplicación para manejar la carga de usuarios durante la subida y descarga de archivos de diferentes tamaños y formatos.
2. **Compresión de archivos:** Se deben probar los tiempos de respuesta y la capacidad de la aplicación para comprimir archivos de diferentes tamaños y formatos en formatos .zip, tar.gz, .rar.
3. **Escenarios de errores:** Se deben probar la capacidad de la aplicación para manejar errores, como archivos corruptos o solicitudes incorrectas.
4. **(20%)** ¿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.

Los parámetros de configuración que se deben considerar para la ejecución de pruebas con JMeter su APM son:

Cantidad de usuarios virtuales (hilos): Es necesario definir la cantidad de hilos de prueba que se usarán para simular la carga de usuarios en la aplicación. Los hilos a utilizar ya fueron definidos en el punto 3.

Duración de la prueba: Se debe definir cuando tiempo se mantendrá la carga de usuarios en la aplicación para sí medir su rendimiento en condiciones de uso continuo

Escenarios de prueba: Se deben definir los diferentes escenarios de prueba que se ejecutarán en la aplicación. Como subida, descarga de archivos, compresión de archivos, entre otros.

Umbrales de aceptación: Se deben definir umbrales para medir el rendimiento de la aplicación en función del tiempo de respuesta y la tasa de errores que pueda tener.

5. **(10%) Defina el Escenario 1.** Se deberá definir un escenario donde se pueda probar cuál es la máxima cantidad de requests HTTP por minuto que soporta la aplicación web. Tenga en cuenta que a futuro realizara las pruebas de estrés con la herramienta Apache Bench (ab) o JMeter. Considere que las pruebas de estrés deberán realizarse desde otros equipos diferentes a los utilizados para ejecutar el servidor web y el servidor de base de datos.

Objetivo: Determinar la máxima cantidad de requests HTTP que puede manejar la aplicación web por minuto sin afectar significativamente el tiempo de respuesta.

Parámetros de prueba:

- Hilos: 15, 50, 100.
- Tamaño del archivo: archivo promedio: (10MB)
- Duración de la prueba: 5 minutos

Configurar JMeter para enviar las solicitudes HTTP a la dirección del servidor del con los parámetros previamente definidos.

Análisis de resultados: A partir de la información obtenida se debe determinar la máxima cantidad de requests que la aplicación puede manejar por minuto. Según los esquemas de demanda definidos. A partir de esta información se deben establecer los umbrales de aceptación para el tiempo de respuesta y la tasa de errores.

Documentación de resultados: Todos los umbrales junto con los parámetros deben estar apoyados en gráficas en función del tiempo. Para tener un mejor entendimiento del funcionamiento de la aplicación bajo estrés.

Las gráficas a utilizar serán:

- **Gráfica de líneas:** Es útil para visualizar la evolución de una métrica a lo largo del tiempo, como por ejemplo la tasa de solicitudes por segundo.
- **Gráfica de barras:** Es útil para comparas las diferentes métricas y umbrales obtenidos en las pruebas de carga.

En las pruebas de estrés el tiempo de respuesta promedio de la aplicación debe ser de máximo 5.000 ms, si este tiempo no se cumple, se concluye que el sistema NO soporta la cantidad de requests de la prueba. En caso de que durante una prueba se generen más de un 5% de errores en los requests de la prueba, se concluye que la aplicación NO soporta la cantidad de requests de la prueba.

6. **(10%) Escenario 2.** Se deberá definir un escenario donde se pueda probar cuál es la máxima cantidad de requests HTTP por minuto que soporta la aplicación web. Tenga en cuenta que a futuro realizara las pruebas de estrés con la herramienta Apache Bench (ab) o JMeter. Considere que las pruebas de estrés deberán realizarse desde otros equipos diferentes a los utilizados para ejecutar el servidor web y el servidor de base de datos.

Objetivo: Determinar el máximo número de procesos asincrónicos que el worker puede manejar sin afectar significativamente el tiempo de respuesta y la tasa de errores.

Parámetros de prueba:

- Hilos: 15, 50, 100.
- Tamaño del archivo: archivo promedio: (10MB)
- Duración de la prueba: 5 minutos

Configurar JMeter para enviar las solicitudes HTTP a la dirección del servidor del worker con los parámetros previamente definidos.

Análisis de resultados: Para esta prueba, se debe aumentar gradualmente el número de procesos asincrónico que se envían al worker. A medida que se aumenta el número de procesos, se deben medir el tiempo de respuesta y la tasa de errores para determinar el punto en el que el worker no puede maneras más procesos asincrónicos.

Documentación de resultados: Todos los umbrales junto con los parámetros deben estar apoyados en gráficas en función del tiempo. Para tener un mejor entendimiento del funcionamiento de la aplicación bajo estrés.

Las gráficas a utilizar serán:

- **Gráfica de líneas:** Es útil para visualizar la evolución de una métrica a lo largo del tiempo, como por ejemplo la tasa de solicitudes por segundo.
- **Gráfica de barras:** Es útil para comparas las diferentes métricas y umbrales obtenidos en las pruebas de carga.

En las pruebas de estrés el tiempo de respuesta promedio de la aplicación debe ser de máximo 8.000 ms, si este tiempo no se cumple, se concluye que el sistema NO soporta la cantidad de procesos asincrónicos de la prueba. En caso de que durante una prueba se generen más de un 5% de errores en los 1 de la prueba, se concluye que la aplicación NO soporta la cantidad de procesos asincrónicos de la prueba.

PRUEBAS DE CARGA RESULTADOS:

Escenario 1:

En este escenario se realizaron pruebas GET de las tareas de un usuario en específico para verificar la capacidad de toda la infraestructura en este caso.

15 hilos en 1 segundo:

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimiento	Kb/sec	Sent KB/sec
Get Tasks	15	2956	2947	3258	3324	3381	2534	3381	0,00%	4,3/sec	7,22	4,02
Total	15	2956	2947	3258	3324	3381	2534	3381	0,00%	4,3/sec	7,22	4,02

50 hilos en 1 segundo:

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimiento	Kb/sec	Sent KB/sec
Get Tasks	50	494	456	653	690	1537	321	1537	0,00%	29,5/sec	49,29	27,41
Total	50	494	456	653	690	1537	321	1537	0,00%	29,5/sec	49,29	27,41

100 hilos en 1 segundo:

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimiento	Kb/sec	Sent KB/sec
Get Tasks	100	1393	1234	1913	2396	5484	355	5910	0,00%	15,3/sec	25,53	14,20
Total	100	1393	1234	1913	2396	5484	355	5910	0,00%	15,3/sec	25,53	14,20

Viendo los resultados anteriores podemos confirmar que las peticiones GET de tareas funcionan con 0% de error en los 3 modos de estrés que fue sometida la aplicación.

Escenario 2:

En este escenario se realizaron pruebas POST de las tareas de un usuario en específico para verificar la capacidad de toda la infraestructura en este caso.

15 hilos en 15 segundos:

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimiento	Kb/sec	Sent KB/sec
Post Files	15	25897	27523	29451	30080	32287	4978	32287	0,00%	22,3/min	0,19	3808,90
Total	15	25897	27523	29451	30080	32287	4978	32287	0,00%	22,3/min	0,19	3808,90

50 hilos en 50 segundos:

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimiento	Kb/sec	Sent KB/sec
Post Files	50	71995	78909	91193	91427	97777	4480	97777	0,00%	23,2/min	0,20	3962,39
Total	50	71995	78909	91193	91427	97777	4480	97777	0,00%	23,2/min	0,20	3962,39

100 hilos en 100 segundos:

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimiento	Kb/sec	Sent KB/sec
Post Files	100	136025	153728	178426	181636	190197	4790	194350	4,00%	23,9/min	0,24	3909,68
Total	100	136025	153728	178426	181636	190197	4790	194350	4,00%	23,9/min	0,24	3909,68

Viendo los resultados anteriores, podemos observar que la aplicación en cuánto a peticiones POST para convertir archivos genera un problema que cuando se mandaron 100 solicitudes en 50 segundos esto generó un 4% de error en las solicitudes. Por lo tanto, la aplicación podría aguantar hasta aproximadamente las 200 solicitudes sin ni siquiera llegar a un 50% de error.