

**Infraestructura computacional**

**Ingeniería de Sistemas y Computación**

**ISIS-2203**

Daniela Mariño - 201412576

Joan Torres - 201315711

María Arévalo - 201415326

**Caso de Estudio 3 – Análisis de Desempeño**

**Logística y Seguridad Aeroportuaria**

1. **Cambios realizados para medir indicadores**

Para medir los indicadores en la clase Cliente se definieron los atributos inicioComunica, inicioConsulta, resultadoTiempoComunicación y resultadoTiempoConsulta. Estos, según el paso en el que se encuentre el protocolo, miden el tiempo actual y final del indicador por transacción.

Para el indicador de tiempo de autenticación, se inicia el contador de tiempo cuando el cliente inicia la comunicación de autenticación del servidor y se finaliza cuando se recibe la respuesta OK del servidor, validando que fue correcta el intercambio de certificados y de llave simétrica.

En el caso del indicador de tiempo de actualización, se inicia el contador cuando el cliente envía la consulta al servidor, y se termina cuando el servidor envía la respuesta respectiva.

Finalmente en la clase Generator, se suman los tiempos de cada transacción y se divide entre el número de consultas del escenario de prueba.

Respecto al número de transacciones perdidas, se tiene una variable donde se cuentan las transacciones exitosas en la clase Generator, y al finalizar la prueba este valor es restado de la carga original.

1. **Identificación de la plataforma**

A continuación se describirá la máquina en la que se corrió el servidor:

Arquitectura: 64-bits

Número de núcleos: 2

Velocidad del procesador: 2,6GHz

Tamaño de la memoria Ram: 16gb

Espacio de memoria asignado a la JVM: Para el consumo de memoria se decidió dejar el espacio de memoria asignado por defecto. El desarrollo de las pruebas no se vió afectado por esto.

1. **Gráficas con seguridad**
2. **Carga fija y #threads vs. Tiempo de autenticación**

A través de estas gráficas se puede comprobar que el hecho de tener mayor cantidad de threads disminuye significativamente los tiempos de autenticación.

1. **Carga fija y #threads vs # de transacciones perdidas**
2. **Carga fija y #threads vs porcentaje de uso de la CPU**
3. **2 threads y tiempo de autenticación vs porcentaje de uso de la CPU por carga**
4. **8 threads y tiempo de autenticación vs porcentaje de uso de la CPU por carga**
5. **Comportamiento de la aplicación ante diferentes niveles de seguridad.**

¿Cuál es el resultado esperado sobre el comportamiento de una aplicación que implemente funciones de seguridad vs. una aplicación que no implementa funciones de seguridad?

La seguridad es un atributo de calidad sumamente importante en las aplicaciones modernas, y aunque su aplicación es esencial sobre cualquier implementacion de codigo, tiene un costo agregado sobre el atributo de calidad relacionado al desempeño, en detalle sobre la latencia de respuesta a las peticiones.

Es decir que sobre un servidor que implementa protocolos de seguridad se espera un mayor tiempo de respuesta entre cada petición, dado que para cada mensaje debe aplicarse un proceso de cifrado y descifrado para dar una respuesta. Así mismo al iniciar una comunicación se asignan threads dedicados a la autenticación de ambas partes (cliente-servidor) en un principio y luego se dedican a la resolución de la petición o mensaje emitido por el cliente.

Así mismo sobre un servidor sin protocolos de seguridad en la autenticación y la confidencialidad del mensaje se corre un mayor riesgo de espionaje, suplantación, o modificación dentro del flujo de mensajes. Al no haber patrones o tácticas en contra de estas amenazas el servidor y el cliente solo se encargan de intercambiar mensajes entre sí, sin necesidad de realizar procesos de cifrado, resumen de hash u otras implementaciones que aumentan el tiempo de respuesta; por este motivo se deduce que se esperan menores tiempos de respuesta entre respuestas de mensajes.

* 1. **Graficas sin seguridad**

1. **Carga fija y #threads vs. Tiempo de autenticación**
2. **Carga fija y #threads vs # de transacciones perdidas**
3. **Carga fija y #threads vs porcentaje de uso de la CPU**
4. **2 threads y tiempo de autenticación vs porcentaje de uso de la CPU por carga**
5. **8 threads y tiempo de autenticación vs porcentaje de uso de la CPU por carga**