Tarea 3: Deployment y modificación HelloWorld

Informe sobre Mediciones de Atributos de Calidad de Rendimiento en Nuestro Sistema Cliente-Servidor

Presentado por:
- **Adri Martinez-A00400842**
- **Johan Guzmán-A00401480**

Introducción

El presente informe documenta la evaluación de los atributos de calidad relacionados con el **rendimiento** del sistema, entendido como la capacidad del software para cumplir sus funciones bajo restricciones de tiempo, precisión y uso de recursos.
Las pruebas se llevaron a cabo en un ambiente controlado de laboratorio, donde **dos equipos actuaron como clientes** y **un equipo como servidor**.

Los clientes se conectaron al servidor a través de la aplicación client.jar, enviando una serie de peticiones que incluían la factorización de números enteros y la ejecución de comandos del sistema (`listifs`, `listports`, `!date`, `!whoami`, `!pwd`, `!ls`, `!ifconfig`, `!ip a`, `!ps -e`, `!free -h`).

Durante las pruebas se capturaron métricas como:

- **ServerTime (ms):** tiempo de procesamiento interno del servidor.
- **RTT (Round Trip Time):** tiempo total de ida y vuelta de la petición hasta la recepción de la respuesta en el cliente.

Estos datos permiten medir 3 de los subatributos de rendimiento: **Throughput, Tiempo de Respuesta y Tasa de Pérdida.**

A continuación, recordamos la definición de los requisitos no funcionales y atributos de calidad, tal como los hemos considerado en nuestro análisis.

Requisitos No Funcionales: Atributos de Calidad

Cada sistema de software implementa un conjunto de funcionalidades, que idealmente corresponden al conjunto de requisitos funcionales solicitados por las partes interesadas. Sin embargo, este conjunto no es la única preocupación del software; las restricciones con las que estas funcionalidades deben ser ejecutadas y entregadas representan la lista de requisitos no funcionales que el sistema de software debe cumplir. Entre estos requisitos no funcionales se encuentran los atributos de calidad del software, como, por ejemplo, el plazo máximo de respuesta a cualquier solicitud. Los atributos de calidad son una preocupación importante en la ingeniería de software porque, a menudo, los sistemas deben ser rediseñados debido a deficiencias en estos atributos.

Los atributos de calidad relevantes para un sistema dado dependen de los requisitos, características y restricciones de cada sistema. Así, lograr un nivel de calidad esperado en cualquier atributo de calidad depende de cómo se implementan y diseñan las funciones del sistema. Lograr los atributos de calidad también depende de la implementación y despliegue del software. Por ejemplo, el rendimiento depende de (i) la comunicación entre componentes, (ii) las funcionalidades de cada componente, (iii) la ubicación de los recursos compartidos, (iv) los algoritmos que implementan las funcionalidades, y (v) la codificación de estos algoritmos. A la luz de esto, existen muchos aspectos a considerar cuando se busca alcanzar un nivel específico de un atributo de calidad.

Rendimiento

El atributo de calidad de rendimiento se define como "El grado en que un sistema o componente cumple con sus funciones designadas dentro de las restricciones dadas, tales como velocidad, precisión o uso de memoria." Este atributo de calidad está relacionado con el tiempo que le toma a un sistema responder a los eventos. Existen muchos tipos de eventos, tales como interrupciones, mensajes, solicitudes de usuarios, entre otros. Los dos aspectos principales que hacen que este atributo de calidad sea complejo son: (i) la cantidad de fuentes de eventos y (ii) los patrones de llegada de los eventos. Ejemplos de fuentes de eventos son el sistema operativo y las aplicaciones de usuario. Los patrones de llegada de eventos pueden caracterizarse como periódicos, esporádicos o estocásticos. Los eventos periódicos tienen un patrón de repetición y posiblemente una fecha límite, mientras que los eventos estocásticos ocurren según una distribución probabilística; los eventos esporádicos son aquellos que no pueden clasificarse como periódicos ni estocásticos.

Debido a sus características, el rendimiento tiene aspectos del sistema que contribuyen a él. Estos aspectos son conocidos como subatributos o factores de rendimiento. Algunos de los subatributos de rendimiento a través de los cuales se puede medir concretamente son los siguientes:

1. **Rendimiento (Throughput)**: La cantidad de trabajo por unidad de tiempo.
2. **Tiempo de Respuesta (Response Time)**: El tiempo transcurrido entre la llegada del evento y la respuesta del sistema (también conocido como latencia), por ejemplo, el tiempo de carga de una aplicación, apertura de pantalla y tiempos de actualización, etc.
3. **Tasa de Pérdida (Missing rate)**: Cantidad de eventos perdidos por unidad de tiempo.

Metodología de Pruebas

Las pruebas se diseñaron para simular un escenario con múltiples clientes concurrentes interactuando con un único servidor.

La configuración fue la siguiente:

- **Entorno:** Tres computadores conectados en la misma red local.
- **Roles:** 1 servidor (`104m19`), 2 clientes (`104m24` y `104m25`).
- **Herramientas:** `client.jar` y `server.jar`.
- **Procedimiento:**
- Los clientes enviaron distintos comandos (cálculo de factorización, comandos del sistema, exploración de interfaces/red, apertura de puertos).
- **Datos de prueba**: Las peticiones enviadas incluyeron:
  - **Factorización de números**: 10, 50, 88, 200, 950, 5000 (Cliente 1); 15, 60, 100, 340, 1000 (Cliente 2).
  - **Comandos del sistema**: `listifs`, `listports`, `!date`, `!whoami`, `!pwd`, `!ls`, `!ifconfig`, `!ip a`, `!ps -e`, `!free -h`.
- Para cada solicitud se midieron **ServerTime** y **RTT**.
- Se compararon los resultados entre clientes.
- Se registraron casos de error o pérdida de conexión (ej. `ConnectionRefusedException`, `TimeoutException`).

Resultados y Fórmulas Utilizadas

Tabla de datos obtenidos

Tipo de Petición	Cliente 1: ServerTime (ms)	Cliente 2: ServerTime (ms)	Cliente 1: ServerTime (rtt)	Cliente 2: ServerTime (rtt)
Números (<100)	41, 1, 1	0, 1, 2	44, 2, 3	2, 3, 4
Números (>100)	4, 10, 437	6, 9, 436	3, 12, 439	7, 11, 438
listifs	10, 0, 0, 1, 0	0, 0, 0, 0, 0	12, 2, 2, 3, 2	2, 2, 2, 2, 2
listports	25, 20, 18, 676	17, 19, 16, 689	27, 25,19, 678	23, 24, 21, 690
Comandos !	40, 17, 13, 15, 19, 17, 45, 17	16, 16, 16, 18, 32, 43, 17, 19	42, 19, 15, 17, 22, 18, 47, 19	19, 18, 16, 18, 21, 34, 45, 19

Tabla 1. Datos obtenidos

A continuación, documentamos cada subatributo de rendimiento teniendo en cuenta la tabla de datos que obtuvimos al ejecutar las consultas, para cada atributo, incluimos:

1. **Definición**: Basada en la descripción proporcionada en tu consulta.
2. **Fórmula**: La ecuación utilizada para calcularlo, adaptada al contexto de tus pruebas (usando ServerTime como tiempo de procesamiento en el servidor, ya que representa el tiempo entre la llegada y la respuesta del sistema).
3. **Medición**: Los valores calculados a partir de los datos, presentados en tablas donde sea efectivo para comparaciones o enumeraciones.
4. **Análisis**: Observaciones breves sobre los resultados en este setup.

Análisis por cada atributo de calidad:

1. **Rendimiento (Throughput)**

   1.1. Definición: Este atributo se refiere a la cantidad de trabajo realizado por el sistema por unidad de tiempo, como el número de transacciones procesadas. Se espera que el sistema pueda manejar un alto volumen de eventos o solicitudes dentro de un periodo determinado, asegurando su eficiencia.

   1.2. Formula: Para calcularlo se hace uso de la formula:

$$\text{Throughput} = \frac{\text{Número total de solicitudes procesadas}}{\text{Tiempo total de ejecución (en segundos)}}$$

   1.3.Medición:

   1.3.1. Cálculo para el Cliente 1

   1. **Conteo de Peticiones**:
      - Números (<100): 3 peticiones
      - Números (>100): 3 peticiones
      - `listifs`: 5 peticiones
      - `listports`: 4 peticiones
      - Comandos `!`: 8 peticiones
      - **Total de peticiones**: 3+3+5+4+8=23
   2. **Suma del Tiempo Total (rtt)**:
      - Números (<100): 44+2+3=49 ms
      - Números (>100): 3+12+439=454 ms
      - `listifs`: 12+2+2+3+2=21 ms
      - `listports`: 27+25+19+678=749 ms
      - Comandos `!`: 42+19+15+17+22+18+47+19=199 ms
      - **Tiempo total (rtt)**: 49+454+21+749+199=1472 ms = **1.472 s**
   3. **Cálculo del Throughput**:

$$ThroughputCliente1 = \frac{23\ peticiones}{1.472\ s} \approx 15.62\ peticiones/s$$

   1.3.2. Cálculo para el Cliente 2

   1. **Conteo de Peticiones**:
      - Números (<100): 3 peticiones
      - Números (>100): 3 peticiones
      - `listifs`: 5 peticiones
      - `listports`: 4 peticiones

- Comandos `!`: 8 peticiones
- **Total de peticiones**: 3+3+5+4+8=23

2. **Suma del Tiempo Total (rtt)**:
   - Números (<100): 2+3+4=9 ms
   - Números (>100): 7+11+438=456 ms
   - `listifs`: 2+2+2+2+2=10 ms
   - `listports`: 23+24+21+690=758 ms
   - Comandos `!`: 19+18+16+18+21+34+45+19=190 ms
   - **Tiempo total (rtt)**: 9+456+10+758+190=1423 ms = **1.423 s**
3. **Cálculo del Throughput**:

$$ThroughputCliente2 = \frac{23\ peticiones}{1.423\ s} \approx 16.16\ peticiones/s$$

1.4. Análisis: Los resultados muestran que el throughput del Cliente 2 fue ligeramente superior al del Cliente 1. Esto se debe a que, a pesar de haber enviado el mismo número de peticiones, el tiempo total de ida y vuelta (RTT) para el Cliente 2 fue menor en la mayoría de los casos, especialmente en operaciones simples como factorización de números pequeños y listifs. Esto podría indicar una menor latencia de red, un canal de comunicación más estable o incluso que el servidor respondió con mayor consistencia durante las pruebas del Cliente 2.

2. Tiempo de Respuesta (Response Time) o Latencia

2.1. Definición: Este atributo se refiere al tiempo que transcurre entre la solicitud de un evento y la respuesta del sistema. Se espera que el sistema responda de manera rápida, minimizando la latencia y asegurando una experiencia de usuario fluida.

2.2. Fórmula:

$$\text{Tiempo de Respuesta Promedio} = \frac{\sum \text{ServerTime de cada solicitud}}{\text{Número de solicitudes}}$$

(Se calcula por comando para mayor detalle).

2.3. Medición:

A continuación, se presentan los cálculos detallados del `Tiempo de Respuesta Promedio` para cada cliente, basados en los valores de `ServerTime` de la tabla de datos que obtuvimos.

Cliente 1

1. **Valores de `ServerTime` (en ms)**:
   - Números (<100): 41, 1, 1
   - Números (>100): 4, 10, 437
   - `listifs`: 10, 0, 0, 1, 0
   - `listports`: 25, 20, 18, 676
   - Comandos `!`: 40, 17, 13, 15, 19, 17, 45, 17
2. **Suma total de `ServerTime`**:

   (41+1+1)+(4+10+437)+(10+0+0+1+0)+(25+20+18+676)+(40+17+13+15+19+17+45+17)=1427 ms

3. **Número total de solicitudes**:

   3 (Números <100)+3 (Números >100)+5 (listifs)+4 (listports)+8 (Comandos !)=23 solicitudes

4. **Tiempo de Respuesta Promedio**:

$$\frac{1427\text{ms}}{23\text{ solicitudes}} \approx 62.04\text{ ms}$$

Cliente 2

1. **Valores de `ServerTime` (en ms)**:
   - Números (<100): 0, 1, 2
   - Números (>100): 6, 9, 436
   - `listifs`: 0, 0, 0, 0, 0
   - `listports`: 17, 19, 16, 689
   - Comandos `!`: 16, 16, 16, 18, 32, 43, 17, 19
2. **Suma total de `ServerTime`**:

   (0+1+2)+(6+9+436)+(0+0+0+0+0)+(17+19+16+689)+(16+16+16+18+32+43+17+19)=1372 ms

3. **Número total de solicitudes**:

   3 (Números <100)+3 (Números >100)+5 (listifs)+4 (listports)+8 (Comandos !)=23 solicitudes

4. **Tiempo de Respuesta Promedio**:

$$\frac{1372\text{ ms}}{23\text{ solicitudes}} \approx 59.65\ ms$$

2.4.Análisis

El tiempo de respuesta promedio para el **Cliente 2 (59.48 ms)** fue ligeramente menor que el del **Cliente 1 (64 ms)**. Esta diferencia, aunque pequeña, sugiere que el servidor tuvo un rendimiento marginalmente mejor o que la carga de trabajo del Cliente 2 fue más eficiente en términos de procesamiento total. Es importante notar que los valores extremos (`437 ms` y `676 ms` para el Cliente 1; `436 ms` y `689 ms` para el Cliente 2) influyen significativamente en el promedio, ocultando el hecho de que la mayoría de las peticiones tienen tiempos de respuesta muy bajos.

3. Tasa de Pérdida (Missing rate)

3.1. Definición: Este atributo se refiere a la cantidad de eventos que el sistema no puede procesar o pierde por unidad de tiempo. Se espera que el sistema minimice la pérdida de eventos para garantizar la precisión y la integridad de la información procesada.

3.2. Fórmula:

$$Missing\ Rate\ = \frac{(Eventos\_enviados\ -\ Eventos\_recibidos)}{T\_total}$$

3.3. Medición:

- **Eventos enviados por cliente:** 23 (para ambos).
- **Eventos recibidos:** 23 (para ambos, ya que no hubo fallos en las respuestas).
- $\boldsymbol{T_{total}}$: corresponde al tiempo total medido por cliente (la suma de todos los ServerTime que ya calculamos antes).
    - Cliente 1 → $T_{total} = 1427$ ms
    - Cliente 2 → $T_{total} = 1372$ ms

3.4. Cálculo:

- **Cliente 1**

$$\text{Missing Rate} = \frac{(23 - 23)}{1427} = \frac{0}{1427} = 0$$

- **Cliente 2**

$$MissingRate = \frac{(23 - 23)}{1372} = \frac{0}{1372} = 0$$

3.5. Análisis:

La tasa de pérdida para Cliente 1 y Cliente 2 es 0, lo que significa que no se perdieron eventos durante las pruebas. Esto implica que todo lo enviado fue procesado y respondido dentro del tiempo total de observación, mostrando que el sistema es confiable en términos de integridad de la comunicación.

Conclusiones

1. Durante nuestras pruebas pudimos comprobar que el sistema respondió de manera correcta a todas las solicitudes enviadas por ambos clientes, sin presentar pérdidas de eventos. Esto nos da confianza en la confiabilidad y estabilidad del sistema.
2. Observamos que, aunque ambos clientes realizaron el mismo número de peticiones, el Cliente 2 obtuvo un rendimiento ligeramente mejor en términos de throughput y tiempo de respuesta. Creemos que esto puede estar relacionado con la conexión de red o con una mayor consistencia del servidor en ese momento.
3. Algo que nos llamó la atención fue la presencia de algunos tiempos de respuesta muy altos (entre 437 ms y 689 ms), que influyeron en el promedio. La mayoría de las solicitudes fueron rápidas, pero estos valores extremos nos muestran que el sistema, aunque en general eficiente, puede tener picos de latencia en operaciones más pesadas.
4. Nos parece importante destacar que no se perdieron eventos durante las pruebas (Missing Rate = 0). Esto confirma que el sistema mantiene la integridad de la comunicación y asegura que toda la información llega al destino.

Recomendaciones

1. Creemos que sería útil realizar más pruebas con un mayor número de clientes conectados al mismo tiempo, para ver cómo se comporta el sistema bajo condiciones de mayor carga.
2. Nos gustaría complementar el análisis con métricas adicionales como la mediana, la desviación estándar o el jitter de los tiempos de respuesta. Esto nos ayudaría a entender mejor la estabilidad y variabilidad del sistema.
3. Consideramos importante revisar los procesos que generan tiempos de respuesta más altos, como el cálculo con números grandes o la exploración de puertos. Si se optimizan estos procesos, se podrían reducir los picos de latencia.
4. Finalmente, creemos que estas métricas pueden servir como una base para planear mejoras futuras en el sistema, pensando en su escalabilidad y en cómo responderá si aumenta el número de usuarios o solicitudes.

Referencias

- ✓ **ISO/IEC 25010:2011.** *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) System and software quality models.* International Organization for Standardization, 2011.