

INTEGRANTES:

- ANGY MARIA HURTADO OSORIO
- DANIEL STIVEN TRUJILLO

INFORME INFORME DE MEDICIÓN DE ATRIBUTOS DE CALIDAD - PERFORMANCE

HELLOWORLD

ENLACE DEL REPOSITORIO <https://github.com/LissaAN1/Tarea3-DeploymentModificaci-nHelloWorld.git>

Introducción

Objetivo del Estudio

Este documento presenta la metodología y resultados de la medición de los atributos de calidad de performance en el sistema cliente-servidor implementado. Se evaluaron tres métricas principales: Latencia, Throughput y Missing Rate.

Sistema bajo Prueba

- Arquitectura: Cliente-Servidor
- Funcionalidades: Procesamiento de comandos (Fibonacci, listifs, listports, ejecución de comandos)
- Escenario: Múltiples clientes enviando solicitudes concurrentes

Metodología de Medición

1. Latencia (Tiempo de Respuesta)

Definición Operacional

Tiempo total desde que el cliente envía una solicitud hasta que recibe completamente la respuesta del servidor.

Instrumentación Implementada

- Se registra el tiempo inicial (requestStartTime) justo antes de enviar la solicitud al servidor.
- Se registra el tiempo final (requestEndTime) inmediatamente después de recibir la respuesta.
- La diferencia entre estos dos tiempos se almacena en una lista (latencias) para su posterior análisis.

```

// Medir latencia
long requestStartTime = System.currentTimeMillis();
totalMessages++;

try {
    response = service.printString(fullMessage);

    // Calcular latencia
    long requestEndTime = System.currentTimeMillis();
    long latency = requestEndTime - requestStartTime;
    latencies.add(latency);
    successfulMessages++;
}

```

Métricas Calculadas

- Latencia promedio: totalLatency / latencies.size()
- Latencia mínima: min(latencies)
- Latencia máxima: max(latencies)

```

// Calcular estadísticas de Latencia (método básico)
if (!latencies.isEmpty()) {
    long minLatency = latencies.get(0);
    long maxLatency = latencies.get(0);
    long totalLatency = 0;

    // Buscar mínimo y máximo
    for (Long latency : latencies) {
        if (latency < minLatency) minLatency = latency;
        if (latency > maxLatency) maxLatency = latency;
        totalLatency += latency;
    }

    double avgLatency = (double) totalLatency / latencies.size();

    System.out.println(" Latencia promedio por solicitud: " + String.format("%.2f", avgLatency) + " ms");
    System.out.println(" Latencia maxima observada: " + maxLatency + " ms");
    System.out.println(" Latencia minima observada: " + minLatency + " ms");
}

```

2. Throughput (Rendimiento)

Definición Operacional

Número de solicitudes procesadas exitosamente por unidad de tiempo (segundos).

Instrumentación Implementada

- Se cuenta el total de mensajes enviados (totalMessages).
- Se cuenta el número de mensajes exitosos (successfulMessages).
- Se registra el tiempo total de sesión (totalTime).

Métricas Calculadas

- Throughput promedio: $\text{totalMessages} / (\text{totalTime} / 1000.0)$

```
// Calcular Throughput promedio (solicitudes/segundo)
double avgThroughput = 0.0;
if (totalTime > 0) {
    avgThroughput = (double) totalMessages / (totalTime / 1000.0);
}
System.out.println("    Throughput promedio: " + String.format("%.2f", avgThroughput) + " solicitudes/segundo");
```

3. Missing Rate (Tasa de Pérdida)

Definición Operacional

Porcentaje de solicitudes que no recibieron respuesta del servidor dentro de un timeout predefinido.

Instrumentación Implementada

- Se cuenta el número de mensajes fallidos (failedMessages).
- Se calcula el porcentaje respecto al total de mensajes enviados.

Métricas Calculadas

- Missing Rate: $(\text{failedMessages} / \text{totalMessages}) * 100.0$