

Informe Técnico del Servidor en C y Pruebas de Carga con JMeter

1. Arquitectura del Servidor

El servidor fue desarrollado en lenguaje C utilizando sockets TCP, con un enfoque de alta concurrencia basado en un pool de hilos (thread pool). Entre sus características principales destacan: - Thread Pool configurable: creación de un número fijo de hilos al inicio para procesar clientes encolados. - Cola de clientes protegida con mutex y variables de condición. - Procesamiento de clientes: recepción de mensajes, validación, conteo de caracteres y respuesta. - Manejo de señales con apagado controlado mediante signalfd. - Estadísticas en tiempo real sobre clientes procesados y activos.

2. Estrategias de Diseño Utilizadas

- Escalabilidad mediante el uso de un thread pool. - Backlog ampliado en listen() para múltiples conexiones concurrentes. - Validaciones seguras y buffers protegidos. - Cierre controlado de sockets tras cada conexión.

3. Resultados de las Pruebas con JMeter

Las pruebas de carga se realizaron con Apache JMeter configurado con 10,000 hilos, un ramp-up de 2 segundos y 10 iteraciones por usuario, resultando en 100,000 peticiones totales. Los resultados principales fueron:

Métrica	Valor
# Samples	100,000
Tiempo promedio de respuesta	~2 ms
Throughput	~4991 peticiones/segundo
Error %	0%
Latencia	0 - 2 ms

4. Razones para Usar C

- Máximo control sobre recursos del sistema. - Rendimiento óptimo sin dependencias de máquinas virtuales. - Escalabilidad con técnicas de concurrencia eficientes. - Ligereza en consumo de CPU y RAM frente a otros lenguajes.

5. Conclusiones

El servidor en C demostró robustez, bajo tiempo de respuesta (~2 ms), alto rendimiento (~5000 peticiones/segundo) y eficiencia en el uso de recursos. El diseño con thread pool y sincronización cuidadosa lo convierte en una solución ideal para aplicaciones críticas con alta concurrencia y baja latencia.

6. Link al repositorio

<https://github.com/AdrianCCRS/tcp-server-c>