

Procesos e Hilos en Sistemas Distribuidos

GÓMEZ EDGAR, Universidad Central de Venezuela, Venezuela

Los sistemas distribuidos basados en el modelo cliente-servidor generalmente necesitan mantener una buena tasa de rendimiento y fiabilidad de respuesta, además de ser capaces de manejar múltiples cargas en concurrencia. Esto es difícil en aplicaciones que son monohilo o que se comportan como tal, debido a la gran cantidad de operaciones que pueden bloquear o demorar el tiempo de respuesta cuando se utiliza solo un hilo principal para procesar todas las peticiones. Esto impacta negativamente la experiencia de usuario con tiempos de espera largos o solicitudes sin respuesta, debido al colapso del proceso principal al encargarse de todas las operaciones una a la vez. Por eso, las operaciones realizadas por procesos se ejecutan en hilos separados, implementando estructuras de datos que soporten concurrencia o utilizando mecanismos de exclusividad para el acceso a los datos compartidos. Esto permite manejar un mayor número de operaciones mientras el hilo principal está listo para delegar nuevos trabajos a otros hilos.

CCS Concepts: • **Organización de sistemas informáticos** → **Sistemas Distribuidos**; *Hilos en software*; *Procesos en software*.

Additional Key Words and Phrases: Sistemas Distribuidos, Arquitectura de Software, procesos en software, hilos en software

ACM Reference Format:

GÓMEZ EDGAR. 2025. Procesos e Hilos en Sistemas Distribuidos. *Proc. ACM Hum.-Comput. Interact.* 1, 1, Article 1 (December 2025), 3 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1. Introducción

Siempre deseamos sistemas más eficientes y, si es posible, que gestionen múltiples operaciones a la vez de manera eficaz. La implementación de multihilos nos brinda la capacidad de manejar diferentes procesos simultáneamente. La arquitectura tradicional, basada en procesos monolíticos, a menudo se revela insuficiente para esta tarea, ya que un solo flujo de ejecución puede convertirse en un punto único de fallo y un cuello de botella para el procesamiento concurrente. La programación multihilo ha emergido como una respuesta fundamental a estas limitaciones. Este documento explora el papel crítico de los procesos y, en particular, de los hilos dentro de los sistemas distribuidos, analizando sus principios, su implementación tanto en clientes como en servidores, y los compromisos necesarios para aprovechar su potencial de concurrencia sin comprometer la estabilidad y eficiencia del sistema.

2. Procesos

El concepto de proceso proviene del ámbito de los sistemas operativos, donde generalmente se define como un programa en ejecución.

Para un sistema distribuido es mejor procesar diferentes tareas usando mayor granularidad (hilos) que hacerlo de manera monolítica, haciendo el sistema capaz de manejar múltiples procesos al mismo tiempo y mejorando el tiempo de respuesta del hilo principal, el cual es el encargado de recibir las peticiones que ejecutarán procesos futuros en hilos separados.

3. Hilos

Un hilo (o thread) se define como una unidad de ejecución ligera que existe dentro de un proceso. A diferencia de los procesos tradicionales, que son pesados, independientes y no comparten memoria, varios hilos pueden coexistir dentro del mismo proceso, compartiendo completamente su espacio de direcciones, recursos abiertos

Author's Contact Information: GÓMEZ EDGAR, Universidad Central de Venezuela, Caracas, Venezuela.

2025. ACM 2573-0142/2025/12-ART1

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

(como archivos y conexiones de red) y el entorno de ejecución. Lo que distingue a cada hilo es su estado de ejecución privado: posee su propio contador de programa, una pila (stack) independiente y un conjunto de registros de CPU. Esto permite que cada hilo ejecute una secuencia de código de manera concurrente con otros hilos del mismo proceso.

La importancia de los hilos en los sistemas distribuidos radica en su granularidad más fina y su bajo costo de administración. Mientras que la creación, conmutación y comunicación entre procesos resulta costosa para el sistema operativo, los hilos permiten manejar múltiples actividades como atender solicitudes de red, realizar operaciones de entrada/salida o procesar datos de manera simultánea dentro de una misma aplicación distribuida, sin incurrir en la sobrecarga asociada a múltiples procesos.

3.1. Hilos en el Cliente

Los hilos en el cliente desempeñan un papel crucial para mejorar la interactividad, el rendimiento y la experiencia del usuario en aplicaciones distribuidas. Un cliente que utiliza múltiples hilos dentro de un mismo proceso puede realizar llamadas a servicios remotos o acceder a recursos de red sin bloquear la interfaz de usuario principal. Un ejemplo de esto son los navegadores web, que hacen una petición de una página web con un hilo diferente al hilo donde corre el programa, evitando que se congele.^{en} operaciones lentas.

La arquitectura multihilo en el cliente también permite manejar de manera eficiente múltiples solicitudes simultáneas a distintos servidores distribuidos. En lugar de procesar las comunicaciones de forma secuencial, el cliente puede lanzar varios hilos, cada uno interactuando con un servicio diferente en paralelo, reduciendo así el tiempo total de ejecución.

3.2. Hilos en el Servidor

Los hilos en el servidor constituyen un modelo fundamental para construir servicios distribuidos concurrentes, escalables y eficientes. La esencia de este enfoque radica en utilizar múltiples hilos dentro de un mismo proceso servidor para atender simultáneamente a varios clientes. Cuando llega una nueva solicitud, por ejemplo, una petición de archivo o una transacción de base de datos, el servidor puede asignarla a un hilo de trabajo independiente, liberando al hilo principal para que continúe aceptando nuevas conexiones. Esto permite que el servidor maneje cientos o miles de clientes concurrentes sin necesidad de crear un proceso pesado para cada uno, optimizando drásticamente el uso de recursos del sistema como la memoria y reduciendo la sobrecarga del cambio de contexto.

La principal ventaja de este modelo es la separación lógica entre la aceptación de conexiones y el procesamiento de las solicitudes. Un hilo puede quedar bloqueado, esperando una lectura de disco o el resultado de una consulta a otro servicio, sin afectar la capacidad de otros hilos para seguir procesando solicitudes.

3.3. Número de hilos en el servidor

Los hilos en el servidor no pueden ser infinitos, ya que existen limitaciones de recursos. En primer lugar, cada hilo ocupa memoria adicional para su pila de ejecución privada, por lo que un número muy elevado puede agotar los recursos de memoria disponibles y comprometer el funcionamiento del sistema. En segundo lugar, y más crítico, es que la ejecución simultánea de numerosos hilos introduce un patrón de acceso a la memoria impredecible y desordenado. Incluso cuando toda la información cabe en la memoria física, el acceso caótico a los datos neutraliza la eficacia de las cachés del procesador, lo que puede hacer que un servidor multihilo rinda peor que uno con un solo hilo.

4. Conclusión

En definitiva, la adopción de un modelo multihilo se presenta como una solución esencial para superar las limitaciones fundamentales de las aplicaciones monohilo en sistemas distribuidos. Al permitir la ejecución

concurrente de múltiples tareas dentro de un mismo proceso, los hilos facilitan un manejo eficiente de numerosas solicitudes simultáneas, mejorando drásticamente la capacidad de respuesta y la experiencia del usuario. Tanto en el cliente, donde se evita el bloqueo de la interfaz durante operaciones de red, como en el servidor, donde se logra una alta escalabilidad para atender a miles de conexiones, el paradigma multihilo optimiza el uso de recursos al ofrecer una granularidad más fina y un costo de conmutación significativamente menor que el de los procesos tradicionales.

No obstante, el diseño multihilo exige una cuidadosa consideración de sus límites prácticos. Un número excesivo de hilos, especialmente en el servidor, puede consumir recursos de memoria críticos e introducir patrones de acceso caóticos a los datos, lo que neutraliza las ventajas de las memorias caché y puede degradar el rendimiento general por debajo incluso del de un sistema monohilo. Por lo tanto, el éxito de su implementación reside en lograr un equilibrio óptimo: aprovechar la concurrencia para la escalabilidad y la capacidad de respuesta, mientras se gestionan estratégicamente los recursos compartidos y se limita el *pool* de hilos para evitar la saturación del sistema. En este balance, el modelo multihilo se consolida como el pilar para construir aplicaciones distribuidas robustas, eficientes y preparadas para las demandas de la computación moderna.

Referencias

- [1] Tanenbaum, A. S. and Van Steen, M. *Distributed Systems: Principles and Paradigms (2nd Edition)*. Pearson Prentice Hall, Upper Saddle River, NJ 07458, 2007.