

Investigación “*Threads*”.

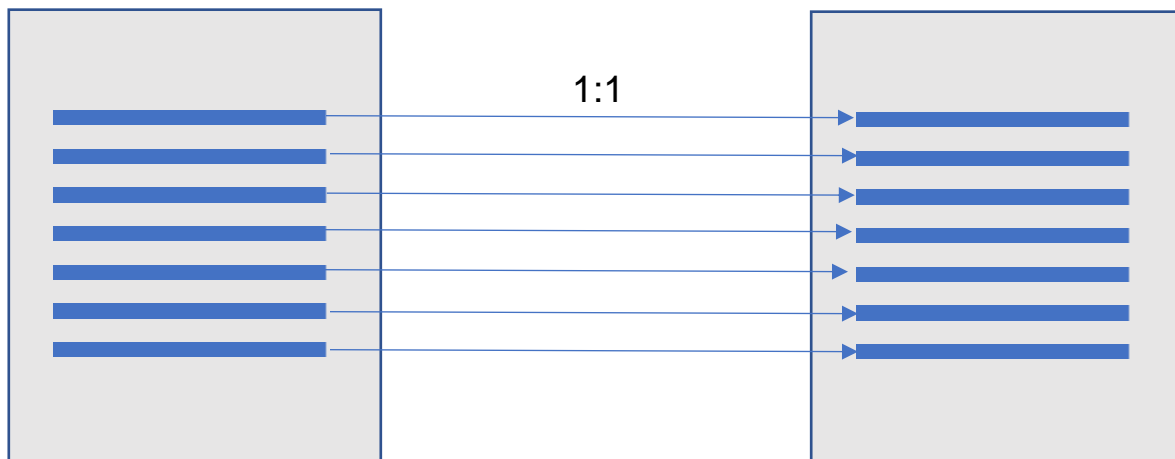
El concepto de concurrencia es la capacidad de un sistema para procesar mas de un hilo al mismo tiempo.

La concurrencia en Java data desde la versión 1.0 donde se introdujeron los “Green Threads” los cuales tenían ciertas características, las cuales no cumplían con las necesidades de la empresa ni la de los desarrolladores. A partir de esas carencias, surgen en la versión 1.1 de Java los hilos de plataforma la cual es la versión actual de hilos de modelo de concurrencia que maneja el lenguaje Java.

Los hilos de plataforma tienen un problema el cual limita al propio lenguaje en cuanto a hilos se refiere y esta es por cada hilo que, de función necesario para nuestra aplicación, se requiere un hilo del sistema operativo a lo cual se le llama “Mapeo uno a uno”

Máquina Virtual de Java

Sistema Operativo

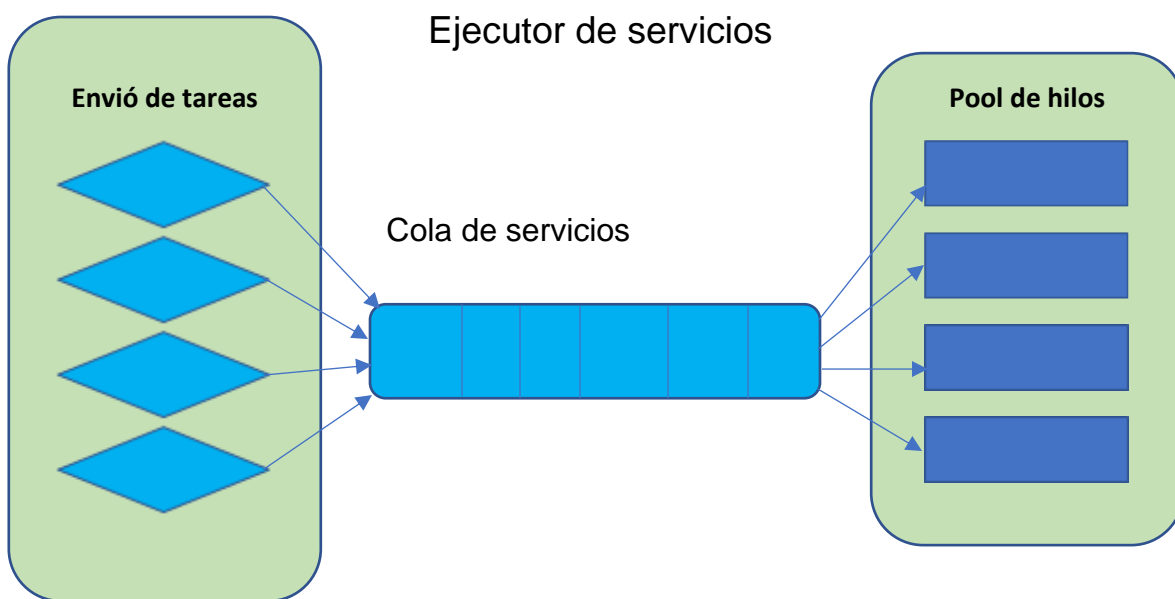


Esto limita la cantidad de hilos que se puede tener a los limites del sistema operativo, si nuestro sistema operativo solo permite 3000 escritores de archivos, entonces nuestra aplicación estará limitada a una menor cantidad de la que provee el sistema debido a que el propio sistema hace uso de estos recursos y sólo dejaría espacio para, por ejemplo, 2000 escritores de archivos, en cuanto esos 2000 escritores de archivos hayan sido utilizados, ya no se puede seguir sacando mas provecho de esta herramienta, además esta aplicación empezará a generar errores.

En una Aplicación real se necesita un hilo por cada solicitud que se hace a esta Aplicación, esto limita a la aplicación, esto hace que en una aplicación Java no se pueda tener tantos procesos ejecutándose simultáneamente como lo permiten otras plataformas.

La creación de hilos anteriormente dicha es bastante costosa en términos de recursos, en vista de este problema, los desarrolladores inventaron un “Pool de hilos”.

Un Pool de hilos es un conjunto de hilos previamente creados listos para ser utilizados, así a la hora que se necesita un hilo para atender una petición no se tiene que crear en el mismo momento, sino que se hace uso de un hilo previamente creado.



Estos se crean cuando se crea una aplicación y mueren cuando muere la aplicación.

Tener estos creados también es bastante costoso, ya que consumen espacio de memoria, aunque no estén en uso.

Si la tarea es muy lenta, entonces hay que esperar a que termine su proceso, si la tarea está detenida entonces el hilo está detenido y el sistema está esperando a que se concluya con la tarea, generando un hilo inutilizable hasta que termine la tarea,

Problemas de los hilos de Java:

- Son costosos de crear y cancelar (threads leaks).
- Están limitados por el sistema operativo.
- Tienen un impacto en el recurso de la Memoria Ram. aproximadamente 1 MB.
- Los hilos de la aplicación pueden llegar a interferir con los hilos del sistema operativo sino se controlan.
- Agregan latencia al funcionamiento de la aplicación.
- Influye en la tasa de transferencia de información de la plataforma.

Impacto

- Problemas para escalar las aplicaciones.
- Incremento de costos por uso de los recursos.
- Carga de responsabilidad a los desarrolladores.
- Pérdida de liderazgo ante nuevas tecnologías y lenguajes como Kotlin, Go, Node, etc.

Actualmente los desarrolladores están trabajando en un proyecto llamado “loom”. Loom es un proyecto que tiene como propósito redefinir el modelo de concurrencia de Java, teniendo este retrocompatibilidad con todas las versiones de Java.