Diseño de modelos de Programación Distribuida

Spry es un modelo de programación para construir aplicaciones que desean intercambiar disponibilidad y consistencia en diversos puntos del código de la aplicación.

Exploramos los desafíos cuando pasamos de la programación secuencial a la programación concurrente.

La construcción de lenguajes para la construcción de aplicaciones distribuidas a gran escala inició en los años de oro;en la decada de 1980 con las innovaciones de la red y la invención del internet.

Para que las aplicaciones distribuidas puedan continuar funcionando cuando no todos los procesos en el sistema pueden comunicarse, como cuando se desarrollan aplicaciones móviles a gran escala o incluso aplicaciones simples donde el estado se almacena en caché en un navegador web; debemos sacrificar el acceso seguro a la memoria compartida.

La mayoría de los modelos de programación que tienen una adopción generalizada en la actualidad son diseñados en estilo von Neumann

Dos de los más conocidos de la época fueron Argus y Emerald, cada uno tomó un enfoque diferente para resolver problemas.

Como reducir la incertidumbre en redes no confiables en el caso de Argus

Y en el caso de Emerald centrada en la movilidad de objetos

La programación distribuida extiende aún más la programación concurrente. Dado un programa que ya es concurrente en su ejecución, los programadores distribuyen los hilos secuenciales de ejecución en múltiples máquinas que se comunican en una red

La programación concurrente es difícil. En el modelo de von Neumann donde las ubicaciones de memoria compartida están mutadas por los hilos secuenciales de ejecución,se debe tener cuidado para evitar el acceso incontrolado a estas ubicaciones de memoria

La programación concurrente extiende la programación secuencial con múltiples hilos de ejecución: esto se hace para aprovechar todo el rendimiento disponible de la computadora permitiendo que se ejecuten varias tareas al mismo tiempo

La mayoría de las aplicaciones distribuidas a gran escala que existen hoy en día se han construido con lenguajes de programación simultáneos como Go, Rust, C / C ++ y Java. Estos lenguajes han adoptado un enfoque de biblioteca para la distribución, adoptando muchas ideas los lenguajes como Emerald y Argus.

Hoy en día, ya sea que esté creando una aplicación web móvil o una aplicación web enriquecedora, los desarrolladores están cada vez más tratando de proporcionar una experiencia "casi nativa" [8], donde los usuarios sienten que la aplicación se está ejecutando en su máquina.

la razón de estos intentos previos de construir lenguajes para los sistemas distribuidos a gran escala es que no pueden capturar los requisitos de los desarrolladores de aplicaciones de hoy