

Simulación Bolsa Concurrente

Informe sobre el problema, la solución y el análisis de rendimiento

Descripción del problema

Este proyecto aborda la simulación de un sistema bursátil simplificado, donde se generan órdenes de compra y venta para un conjunto de acciones. El objetivo es procesar estas órdenes de manera simultánea y mostrar en tiempo real cómo se van ejecutando.

Solución implementada

1. **Generación de órdenes (Productor):** Se crea un proceso productor que, de manera aleatoria, genera órdenes de compra o venta. Cada orden contiene un identificador, el ticker de la acción, un precio ajustado (según sea compra o venta), la cantidad, y un timestamp.
2. **Procesamiento de órdenes (Consumidores):** Se crean varios procesos consumidores, cada uno con un límite máximo de órdenes que puede procesar. Estos consumidores toman las órdenes de la cola compartida y realizan el matching:
 - Orden de compra vs. lista de órdenes de venta pendientes.
 - Orden de venta vs. lista de órdenes de compra pendientes.Se usa un **Lock** para asegurar que el matching se realice de manera atómica, evitando condiciones de carrera.
3. **Interfaz gráfica (Tkinter):** Muestra los eventos en tiempo real, incluyendo la generación de órdenes, el procesamiento y los matches realizados.

Análisis de rendimiento

- **Escalabilidad:** El uso de **multiprocessing** permite aprovechar varios núcleos de CPU, mejorando el rendimiento frente a un único proceso. Sin embargo, la comunicación y sincronización mediante colas y locks añade sobrecarga.
- **Sección Crítica:** El mayor coste se concentra en la sección de matching, protegida por un **Lock**. Al aumentar demasiado el número de consumidores, podría incrementarse la contención en esta sección.
- **Simulación:** Se escogió un número aleatorio de órdenes y consumidores para ofrecer variabilidad. Con un mayor volumen de órdenes y más consumidores, el rendimiento sigue siendo aceptable gracias a la concurrencia, aunque podría requerir ajustes adicionales para optimizar el uso del lock.

Simulación Bolsa Concurrente

Documento explicativo

Este programa simula una mini bolsa de valores con Python. Primero implementé un productor que fabrica órdenes de compra o venta para diferentes acciones listadas, y luego unos consumidores que se encargan de revisar esas órdenes y buscar quién compra y quién vende.

Usé varios procesos, cada uno con una parte del trabajo. Así, mientras uno genera órdenes, otros las procesan y, de esa manera, el programa es más eficiente que si todo estuviera en un solo proceso.

Para que no haya problemas cuando se comparan órdenes (por ejemplo, que dos procesos quieran actualizar la misma lista al mismo tiempo), añadí un Lock. Así, cuando un consumidor entra a buscar las órdenes, los demás tienen que esperar a que termine para no pisarse.

Por último, para mejorar la visualización del programa, implementé una ventana usando Tkinter para que se vea en vivo cómo se generan las órdenes y se hacen los cruces de compra y venta. Cada evento se muestra con un color distinto para que sea más fácil entender lo que está pasando.