

Programación Concurrente

Trabajo Práctico

Entrega: 31 de Mayo de 2023

Se desea realizar una evaluación empírica de distintas implementaciones de un conjunto concurrente sobre listas enlazadas. Para ello se deberán evaluar los tiempos de ejecución en distintos escenarios en donde varios hilos realizan concurrentemente distintas operaciones sobre una estructura de datos compartida.

1. Como estructura general de los escenarios puede considerar que:
 - un escenario consiste de una cantidad fija de hilos que se crean al inicio del programa y comparten una única estructura de datos concurrente.
 - todos los hilos realizarán la misma cantidad fija de operaciones.
 - cada hilo realizará un sólo tipo de operación, pero distintos hilos pueden diferir en las operaciones que ejecutan.
2. Se deberá analizar cómo varía el tiempo de ejecución respecto de:
 - la variación en la proporción de hilos que ejecutan determinadas operaciones, manteniendo constante la cantidad de hilos totales.
 - la variación de la cantidad de hilos totales, si se preserva el número total de operaciones.
 - la variación de la cantidad de hilos totales, si se mantiene constante la cantidad de operaciones que ejecuta cada hilo.

Para paliar la variabilidad en los resultados debido a aspectos no controlables, se deberá replicar la ejecución de cada escenario.

3. Se solicita:
 - a) Implementar en Java las siguientes versiones de conjuntos sobre listas: (i) locks de granularidad fina, (ii) sincronización optimista y (iii) sin locks.
 - b) Definir escenarios.
 - c) Implementar en Java las clases que permitan ejecutar escenarios y tomar medidas sobre el tiempo de ejecución.
 - d) Ejecutar los experimentos y recoger resultados sobre el tiempo de ejecución.
 - e) Analizar los resultado obtenidos.

1. Condiciones de Entrega

El trabajo se realizará en grupos de 2 (dos) integrantes. *Configuramos una respuesta automática "No" para cualquier pedido de excepción.*

La entrega deberá incluir:

- Código fuente. Esto incluye herramientas (como scripts, hoja de cálculos) usadas para obtener las métricas relevantes.
- Logs de salidas de la ejecución.
- Reporte que incluya una descripción de la/s estrategia/s utilizadas para definir escenarios, el análisis de los resultados y una descripción del entorno donde se corrieron los experimentos.

Notas:

- Dado que implementaciones de estos algoritmos podrían obtenerse a partir de repositorios públicos y/o herramientas de generación de texto, la evaluación podría aplicar medidas de distancia entre soluciones como parte del criterio de aprobación.
- La entrega puede hacerse enviando un email a hmelgra@dc.uba.ar conteniendo un único archivo .zip o un link a un repo.

2. Bibliografía

- Martin Sulzmann, Edmund S.L. Lam y Simon Marlow. “Comparing the Performance of Concurrent Linked-list Implementations in Haskell”. En: SIGPLAN Not. 44.5 (oct. de 2009), pags. 11-20. issn: 0362-1340. doi: 10.1145/1629635.1629643. url: <http://doi.acm.org/10.1145/1629635.1629643>.
- Rodrigo Medeiros Duarte y col. “Concurrent Hash Tables for Haskell”. En: Programming Languages. Ed. por Fernando Castor y Yu David Liu. Cham: Springer International Publishing, 2016, pags. 110-124. isbn: 978-3-319-45279-1.