JAVATAR 

Computación Concurrente

Segunda tarea

Team Javatar:
Carlos Cruz Rangel
Victor Hugo Gallegos Mota
Jose Demian Jimenez Salgado
Ernesto Muñoz Nieves
Juan Carlos Bautista Sandoval

September 15, 2022

Table of Contents

1 Introducción

Zeratun estaba muy contento de que pudo ganar el dinero para poder reparar sus puertas, pero mientras iba de viaje del Home Depot a su casa junto a su amigo Blue Devil, unos asaltantes los acorralaron, les dijeron que se veían muy listos por lo que no se llevarían sus rines si contestaban bien las siguientes preguntas, Zeratun al no tener otra opción accedió Así mismo también les dejaron un par de ejercicios los cuales deben solucionar correctamente, pues ellos son amantes de la computación, pero no son muy buenos aun en cosas concurrentes, por lo que le dice lo siguiente:

2 Preguntas

2.1 Explica con tus palabras la diferencia entre un Hilo y un Proceso

RESPUESTA: Los hilos son entidades de ejecución independiente que viven dentro de los procesos y, por tanto, viven dentro del mismo espacio de direcciones de memoria que otros hilos, lo que permite acceder a cualquier dato dentro del mismo proceso, los procesos son entidades independientes para el SO (tienen su propia memoria, código, montículo, etc), mientras que los hilos comparten memoria, código, montículo, etc. En pocas palabras un hilo es un proceso ligero y un proceso es un programa en ejecución.

2.2 ¿Para que sirve el método Join?

RESPUESTA El método join suele utilizarse para mantener un orden en la secuencia de los hilos. Así, podemos arrancar una secuencia de hilos llamando a join para que cada uno finalice en el orden que hemos marcado según la llamada a join.

2.3 ¿Qué pasa si no le hacemos Join a los hilos?

RESPUESTA: Cuando un hilo queda bloqueado liberará el bloqueo para que otro hilo pueda entrar en la sección crítica del objeto y desbloquearlo, puede ser necesario esperar a que un determinado hilo haya finalizado su tarea para continuar, pero como no se ha llamado a join, no se podrá manejar el orden de los hilos causando que entren en la región critica de una manera no deseada.

2.4 Da dos ejemplos en la vida real y dos ejemplos en Computación o Programación donde se puedan ejemplificar los siguientes conceptos:Concurrencia, Paralelizable, Concurrencia Paralelizable.

RESPUESTA Concurrencia:

EJEMPLOS EN PROGRAMACIÓN:

En los video juegos o se usará para la recuperación de claves, la interfaz gráfica y el procesamiento de datos recibidos por el servidor para juegos en línea, sin mencionar si es un juego cooperativo donde se utilizan hilos para toda la funcionalidad.

En los servicios web donde el servidor tendrá que lidiar con muchas solicitudes al mismo tiempo. En este contexto, se vuelve interesante tener múltiples hilos de ejecución, permitiendo que cada uno de ellos atienda una solicitud.

EJEMPLOS DE VIDA COTIDIANA:

Tenemos unos juegos olímpicos y nos enfocaremos en las variantes de natación como nado sincronizado, clavados, y 100 metros y tenemos varias albercas para el evento, para que cada competencia se lleve acabo y no hacerlo de una manera secuencial, la competencia de clavados de hace en la alberca A a las 10 am, la de nado sincronizado se hace a la 10:30 am en la alberca B y por ultimo los 100 metros se hacen a las 10 am en la alberca C

Tenemos a un estudiante el cual esta haciendo su tarea pero tiene necesidades por lo que interrumpe su tarea para ir a comer y luego continuar con la tarea, luego la vuelve a interrumpir para escuchar su clase en linea y finalmente vuelve para terminar la tarea.

Paralelizable:

EJEMPLOS EN PROGRAMACIÓN:

Al descargar música en spotify, descargas un número determinado de canciones al mismo tiempo, cada canción es independiente de la otra, por lo que la velocidad y el tiempo que tarda en descargarse cada una no afectará al resto de canciones incluso puedes escuchar alguna canción mientras se descarga.

Las paginas que se encargan de dar recomendaciones de precios de diferentes paginas como cuando buscas un celular te dice cuanto cuesta en walmart, amazon, ... etc, tiene hilos para visualizar el precio en estas paginas

EJEMPLOS DE VIDA COTIDIANA:

Puede ser si hay dos contadores de efectivo, la tarea se puede dividir de manera efectiva y los clientes podrán formar dos colas para poder realizar sus pagos dos veces más rápido

Los deportes sincronizados como nado, gimnasia, ballet.

Concurrencia Paralelizable:

EJEMPLOS EN PROGRAMACIÓN:

Supongamos un programa que realiza dos operaciones aritméticas, tales son y=3 y x=3, como bien se pueden hacer concurrentes por que no importa cual de ellas se ejecuta primero, pero también se puede palelizar ya que no tiene una zona critica por que se pueden hacer exactamente al mismo tiempo.

Supongamos un dron que tiene que observar el recorrido y esquivar obstáculos, como se observa tiene que ser un programa concurrente ya que realiza varias acciones, y se puede paralelizar ya que una parte puede determinar si esquiva el obstáculo y otra parte puede encargarse de la visualización del camino.

EJEMPLOS DE VIDA COTIDIANA:

Tenemos a 5 programadores y cada uno tiene que hacer tareas y pude ser de manera concurrente haciendo un repo de github y cada uno trabajando con los cambios que se van haciendo al programa, pero se puede paralelizar haciendolo desde un editor de código en tiempo real y colaborativo y todos trabajando al mismo tiempo.

Tenemos una competencia de football de seis equipos y para saber cual equipo es el ganador y pasar a la siguiente ronda de eliminación se hace de manera concurrente los partidos, es decir, uno empieza a las 2 en el estadio A otro a las 2:30 en el estadio B y otro a las 3:30 en el estadio C, como podemos notar esto se puede paralelizar si todos empiezan a la misma hora en su respectivo estadio.

3 Primer problema

Nuestro amigo el Inge acaba de paralelizar un código para su chamba, con la finalidad de que se ejecute más rápido. Por el momento, solo cuenta con una computadora con 2 poderosos núcleos, la cual produjo un SpeedUp S2. El Inge quiere saber cuántos núcleos adicionales tendría que comprar para alcanzar el mejor desempeño posible.

3.1 Inciso A

Ayuda a su amigo el Inge y utiliza la Ley de Amdahl para derivarle una fórmula Sn (SpeedUp con n procesadores) en términos de n y S2.

RESPUESTA

$$lim_{2\to\infty}\frac{1}{(1-p)+\frac{p}{2}}$$

3.2 Inciso B

Nos volvemos a encontrar al Inge y te dice triste que uno de sus compañeros logro un ascenso en la empresa, pero cree que es falso lo que hizo, pues te cuenta lo siguiente: “Logre optimizar el programa del Inge un 10x haciendo únicamente el 35% de su código paralelo.” El Inge siente que miente pero no sabe como demostrar la mentira por lo que nos pidió ayuda, ¿Lo que dice su compañero de trabajo es verdad?

RESPUESTA

Utilizando la **ley de Amdahl** llegamos a la conclusión, que si se paralelizo solamente el **35%** de su código entonces el speedup máximo es el siguiente :

$$lim_{x\to\infty}\frac{1}{(1-0.35)+\frac{0.35}{n}} = \textbf{1.5384x}$$

por lo tanto lo que dice su compañero de trabajo es completamente **falso**.

4 Segundo problema

Ayudanos usando la ley de Amdahl para resolver lo siguiente:

4.1 Inciso A

Tenemos un programa con un método M que no podemos paralelizar de ninguna manera, lamentablemente este metodo es un metodazo, pues cuenta con el 45% del tiempo de ejecución del programa . ¿Cuál seria el LIMITE de speedup que se puede lograr ejecutando el programa en una máquina con n procesadores? (Expresar solamente)

RESPUESTA Tenemos un método M de un programa de este método se esta usando el 45% del tiempo de ejecución por lo que este porcentaje no se puede paralelizar entonces queda un 55% del tiempo de ejecución el cual si se puede paralelizar entonces usando la formula del speedup:

$$\left[\frac{1}{(1-.55)+(\frac{.55}{n})}\right]$$

para lo del limite el máximo vendría siendo cuando n tiende al infinito entonces nuestra formula quedaria asi:

$$lim_{x\to\infty} \frac{1}{(1-0.55)+\frac{0.55}{n}}$$

4.2 Inciso B

Ahora supón que M representa el 30% del tiempo de ejecución del programa, ¿Cuál sería el SpeedUp Máximo que podría alcanzar nuestro programa si el número de procesadores NO estuviera LIMITADO?

RESPUESTA

Utilizando la **ley de Amdahl** llegamos a la conclusión, que El límite cuando n tiende a infinito de :

$$lim_{x\to\infty} \frac{1}{(1-0.30)+\frac{0.30}{n}} = 1.4285x$$

por lo tanto la velocidad o speed Up sería **1.4285**.

5 Tercer problema

Tenemos que calcular el SpeedUp que tendría un programa con 100 procesadores, si al medirlo en forma secuencial se tarda 188 segundos y con 2 procesadores se tarda 104 segundos.

RESPUESTA

$$Speedup = \frac{tiempo\ ejecucin\ secuencial}{tiempo\ ejecucin\ con\ mejora} = \frac{188}{104} = 1.80769230769$$

6 EXTRA

Tenemos un programa que resuelve una tarea secuencialmente en $n^3 * Log10(n)$ con una unidad de tiempo de 2 nanosegundos (10-9 s). De esta manera, si $n = 1000$, el computo tomaría $1000^3 * Log10(1000) * 2ns = 6segundos$. Supón que creamos un algoritmo concurrente que trabaja de manera completamente eficiente, es decir, el cómputo total tarda $n^3 Log10(n)/p$, donde p es el numero de hilos. ¿Qué tan grande es la entrada que puede manejar en 1 segundo, 1 minuto y 1 semana? Prueba con 1 hilo, 8 hilos, 1000 hilos y 1000000 hilos.

RESPUESTA:

Veamos que un segundo tiene 1000000000 nanosegundos, un minuto tiene 60000000000 nanosegundos y una semana tiene 604800016558522 nanosegundos.

Table 1: Resultados obtenidos.

# de hilos	**1 segundo**	**1 minuto**	**1 semana**
1	≈ 566.307	≈ 2083.133	≈ 40342.26
8	≈ 1095.825	≈ 4051.929	≈ 79047.41
1000	≈ 5127.0839	≈ 19135.094	≈ 379471.83
1000000	≈ 47463.893	≈ 178755.08	≈ 3586549.58

7 Referencias

Oaks, S., & Wong, H. (2004). Java Threads: Understanding and Mastering Concurrent Programming (3.a ed.). O'Reilly Media.

Raynal, M. (2015). Concurrent Programming: Algorithms, Principles, and Foundations (2013 ed.). Springer.

Wellings, A. J. (2004). Concurrent and Real-Time Programming in Java (1.a ed.). John Wiley & Sons Inc.

Méndez, P. J. T. (2006). PROGRAMACIÓN CONCURRENTE. Paraninfo.