

Actividad Integradora 5.3 Proyecto Paralelo

**TC2037 - Implementación de Métodos Computacionales**

Profesor: Luis Humberto González Guerra

8 de junio de 2022

Integrantes del Equipo:

Jose Oscar Cazares - A00826747

Andrés Aguirre Rodríguez - A01284373

Alberto Horacio Orozco Ramos - A00831719

**Reflexión de Oscar**

Para nuestro proyecto, se utilizó el lenguaje de programación de Erlang, que es conocido como un lenguaje funcional, que, además, cuenta con la habilidad de hacer múltiples procesos (conurrencia). En sí, lo que se realizó primero en nuestro programa fue una función secuencial. Recorría un documento de texto en donde se lee línea por línea. Cada línea se separaba en una lista. Un ejemplo sería la siguiente línea: "1 + 2". Esto, dentro de una lista se separaría de la siguiente manera: ["1", "+", "2"]. El programa después convierte los números string a tipo número entero, después se analiza el segundo elemento de la lista, que sería el operador, y se realiza la operación correspondiente. Un archivo de salida guarda el resultado de las operaciones. Cuatro archivos se procesan de manera secuencial; dicho proceso llama la función una por una hasta terminar. La manera concurrente hace el proceso simultáneamente (procesa los cuatro archivos a la misma vez), y, por consecuencia, el resultado (output) se genera de una manera más rápida.

Definitivamente, en casos en donde se procesa mucha información (área de Big Data), es muy necesario utilizar todos los núcleos de las computadoras, sino una simple versión secuencial de un programa tomaría mucho tiempo en ejecutar. Con respecto a la ética, esperemos que las personas que utilicen esta tecnología lo utilicen para el beneficio de la sociedad y no hacer programas maliciosos Ejemplos: robo de datos financieros, datos personales, datos de vuelos, etcétera.

**Reflexión de Andrés**

En el caso de esta actividad integradora, se nos pidió un programa que procesa archivos de manera secuencial y concurrente. Después de programar ambas versiones es evidente el hecho de que el programa secuencial es más lento que el concurrente, ya que en lugar de manejar varios trabajos a la vez, solo hace uno. Contrario a esto, el concurrente hace uso de los núcleos varios del procesador donde se está usando y termina el proceso de manera más eficaz.

En mi opinión al final del día la elección del tipo de programa que se debe de usar depende de qué clase de problema quieras resolver o cuantas cosas quieras procesar a la vez, si se quiere un programa que procese una cosa a la vez y deje más núcleos para otros procesos se puede usar el método secuencial, pero si se necesita más eficacia es mucho mejor usar el método concurrente, y al final del día el programa concurrente se deriva del secuencial, significando que no conlleva mucho más trabajo su implementación.

**Reflexión de Alberto**

Como reflexión final para este proyecto, considero que en general la programación paralela es un recurso bastante óptimo y de mucho valor para proporcionar más agilidad, rapidez y reducciones de tiempo muy significativas, ayudando en su gran mayoría al procesamiento de datos mediante métodos que en situaciones secuenciales consumen demasiados recursos de la computadora. En esencia, tanto la programación concurrente y la paralela conlleva muchos beneficios y utiliza los procesadores de los computadores de una forma muy perspicaz, obteniendo un resultado bastante conveniente para hacer rendir, por ejemplo, a estructuras de datos bastante complejas que contienen procesos bastante elaborados que en la programación secuencial tardaría bastante tiempo y recursos obtener el mismo resultado.

Esta actividad en específico se utilizaron estos conceptos y los plasma de una forma clara y concisa, aunque no podemos notar una diferencia muy marcada entre el tiempo de velocidad de respuesta del programa secuencial con respecto al tiempo del paralelo; esto debido a que no se están ejecutando procesos muy largos o bien iterando demasiadas veces, por lo que podríamos pensar que este recurso podría ser más útil en un contexto demasiado exigente en cuanto a la rapidez de respuesta y eficiencia de un algoritmo que tenga la necesidad de atender y resolver múltiples procesos al mismo tiempo, los cuáles un algoritmo convencional (secuencial) no podría satisfacer de ninguna forma. Debemos adaptarnos a las necesidades que existen hoy en día para otorgar productos que solicitan programas que resuelvan cosas de forma múltiple y al instante, y qué mejor solución que elaborar algoritmos de forma concurrente o paralela que ahorren líneas de código, tiempos de espera, procesamiento y recursos de un mismo sistema.