

Proyecto Final

Curso de Sistemas Operativos y Laboratorio

Programación paralela con algoritmo genético de optimización

Miembros del equipo

Juan Pablo Bedoya Sánchez
José Manuel González Grisales
Jonatan Jair Leal González

Resumen

El presente proyecto busca implementar y comparar el rendimiento de un algoritmo genético en sus versiones secuencial y paralela, evaluando su eficiencia en diferentes lenguajes de programación y variando la cantidad de núcleos utilizados en el procesamiento. Dado que los algoritmos genéticos son ampliamente utilizados para resolver problemas de optimización compleja, su paralelización ofrece una oportunidad significativa para mejorar tiempos de ejecución y reducir el consumo de recursos.

La metodología del proyecto incluye la implementación del algoritmo en múltiples lenguajes, la ejecución en condiciones controladas, y la medición del rendimiento con el uso de técnicas estadísticas para validar los resultados obtenidos. Se realizarán análisis de speedup y eficiencia para determinar el impacto de la paralelización, garantizando que los resultados no solo sean más rápidos, sino también consistentes y precisos en comparación con las versiones secuenciales.

Este estudio permitirá evaluar la viabilidad de aplicar algoritmos genéticos paralelizados en distintas condiciones, esperando obtener conclusiones respecto a las diferentes métricas propuestas en el estudio a realizar.

Introducción

En la actualidad, muchos problemas de optimización compleja, como la planificación de rutas, el diseño de redes, o la asignación de recursos, no pueden resolverse de manera eficiente utilizando métodos exactos tradicionales debido a la inmensidad del espacio de búsqueda. Los algoritmos genéticos han demostrado ser una herramienta poderosa para abordar este tipo de problemas, ya que ofrecen soluciones aproximadas en un tiempo razonable al simular procesos evolutivos. Sin embargo, la implementación de estos algoritmos puede ser altamente costosa en términos de tiempo de ejecución y consumo de recursos, especialmente cuando se aplican a problemas de gran escala o alta dimensionalidad.

El desafío que aborda este proyecto radica en analizar el rendimiento de los algoritmos genéticos mediante su paralelización, distribuyendo las cargas de trabajo entre múltiples núcleos de procesamiento. En la actualidad, la creciente disponibilidad de procesadores multinúcleo y sistemas paralelos en el ámbito tecnológico ofrece una oportunidad significativa para acelerar la ejecución de estos algoritmos. Sin embargo, existe la necesidad de evaluar si la paralelización realmente se traduce en beneficios sustanciales en términos de velocidad y eficiencia comparada con las versiones secuenciales.

El desarrollo de este desafío es crucial en el contexto tecnológico actual, donde la resolución rápida y eficiente de problemas complejos es clave para la competitividad en áreas como la inteligencia artificial y la ciencia de datos. La optimización de recursos computacionales permite reducir tiempos de procesamiento y maximizar el rendimiento, especialmente en entornos de computación en la nube. Este proyecto ayudará a identificar las mejores prácticas para implementar algoritmos genéticos paralelizados en diversos lenguajes, respondiendo a los desafíos de la computación moderna.

Antecedentes o marco teórico

Un algoritmo genético (**AG**) es un método para solucionar problemas de optimización con o sin restricciones basándose en un proceso de selección natural que imita la evolución biológica. Este algoritmo modifica repetidamente una población de soluciones individuales. En cada paso, el algoritmo genético selecciona individuos de la población actual aleatoriamente y los utiliza como padres para producir los hijos de la siguiente generación. Tras varias generaciones sucesivas, la población "evoluciona" hacia una solución óptima. [1]

La computación paralela consiste en el uso simultáneo de múltiples procesadores o núcleos que ejecutan cada uno una serie de instrucciones que conforman las distintas partes en las que se ha descompuesto un problema computacional para resolver.

Para poder poner en marcha el cómputo paralelo, el problema computacional debe dividirse en distintos componentes, trabajos o problemas que puedan ser resueltos al mismo tiempo, las instrucciones de estos se deben poder ejecutar en cualquier momento y debe ser posible resolver los problemas cada vez en menos tiempo cuantos más recursos informáticos estén trabajando a la vez.

Los recursos informáticos que se utilizan en el procesamiento en paralelo son o una computadora con múltiples procesadores/núcleos o múltiples computadoras conectadas en red (computación distribuida). [2]

Respecto a la relación con los sistemas operativos, se encuentran los siguientes componentes teóricos y prácticos:

1. Manejo de Procesos e Hilos:

En los algoritmos genéticos, cuando se aplica la computación paralela, es fundamental dividir las tareas en procesos o hilos que puedan ejecutarse simultáneamente. La paralelización de un algoritmo genético requerirá una comprensión profunda del modelo de hilos del sistema operativo, para aprovechar al máximo los núcleos disponibles y evitar la sobrecarga de la CPU.

2. Planificación del Procesador:

Los sistemas operativos también se encargan de la planificación del procesador, decidiendo cómo distribuir el tiempo de CPU entre los diferentes hilos o procesos. La computación paralela en un algoritmo genético se beneficia de una planificación eficiente, ya que se deben distribuir los trabajos de manera que se minimicen los tiempos de espera y se maximice la utilización de los núcleos de procesamiento.

3. Memoria Compartida y Comunicación entre Procesos:

Los sistemas operativos proporcionan los mecanismos necesarios para la comunicación entre procesos, ya sea mediante memoria compartida, pipes o colas de mensajes, lo cual es esencial cuando se implementa un algoritmo genético paralelo. Los distintos hilos o procesos que ejecutan diferentes partes del algoritmo deben compartir información, como la población actual de soluciones o los resultados de evaluaciones intermedias, de forma segura y eficiente.

4. Manejo de Recursos y Eficiencia:

Finalmente, los sistemas operativos también juegan un papel crucial en la gestión de recursos. En la ejecución de algoritmos paralelos, es vital que los recursos, como la memoria, la CPU y el almacenamiento, sean administrados eficientemente. El estudio de Sistemas Operativos te permite comprender cómo los recursos del sistema pueden ser gestionados para optimizar el rendimiento de un algoritmo genético paralelo.

La relación entre la computación paralela, los algoritmos genéticos y los temas del curso de Sistemas Operativos es directa. El éxito de la implementación de estos algoritmos en un entorno paralelo depende en gran medida de la correcta gestión de procesos, hilos, memoria y recursos que el sistema operativo facilita mediante las APIs que expone.

Objetivos

PRINCIPAL:

- Implementar y comparar un algoritmo de optimización genética tanto en su versión secuencial como paralela, evaluando su rendimiento en distintos lenguajes de programación.

ESPECÍFICOS:

- Desarrollar e implementar el algoritmo genético en diferentes lenguajes de programación, considerando tanto su versión secuencial como su versión paralela.
- Experimentar con la paralelización del algoritmo utilizando distintas cantidades de núcleos y evaluar el impacto de esta configuración en el rendimiento.
- Realizar una comparación detallada entre las versiones secuenciales y paralelas, midiendo métricas clave como tiempo de ejecución y consumo de recursos.
- Aplicar métodos estadísticos rigurosos para analizar los resultados y validar la significancia de las mejoras en el rendimiento de la versión paralela.
- Presentar los resultados obtenidos a través de un análisis estadístico.

Metodología

El desarrollo de este proyecto se basará en una investigación exhaustiva sobre la implementación de algoritmos genéticos de optimización, con un enfoque particular en las versiones secuenciales y paralelas en distintos lenguajes de programación. Para ello, se seguirán los siguientes pasos:

- **Revisión bibliográfica y técnica:** Se realizará una búsqueda de fuentes académicas y técnicas que proporcionen información sobre la implementación eficiente de algoritmos genéticos. Esto incluirá estudios sobre optimización en diferentes entornos de programación y las metodologías más recientes para la paralelización de estos algoritmos.
- **Selección del lenguaje y entornos de programación:** Se investigarán varios lenguajes de programación, evaluando tanto su capacidad para implementar algoritmos secuenciales como su soporte para técnicas de paralelización. Se seleccionarán los lenguajes más apropiados con base en criterios como rendimiento, facilidad de paralelización y soporte de bibliotecas especializadas.
- **Diseño e implementación del algoritmo:** Una vez seleccionado el algoritmo genético y los lenguajes de programación, se procederá a su implementación tanto en su versión secuencial como en su versión paralela. Esta etapa será un esfuerzo colaborativo entre los miembros del equipo.
- **Ejecución y obtención de resultados:** Se ejecutarán las versiones secuencial y paralela del algoritmo en diferentes condiciones experimentales, recopilando datos clave como el tiempo de ejecución y consumo de recursos..
- **Análisis estadístico de los resultados:** Para evaluar el rendimiento de las distintas versiones del algoritmo, se aplicarán métodos estadísticos apropiados.
- **Conclusiones:** A partir del análisis de los datos obtenidos, se elaborarán conclusiones sobre la viabilidad y eficacia de la paralelización del algoritmo genético en los lenguajes seleccionados.

Diseño de Experimentos

El experimento tiene como objetivo principal evaluar y comparar el rendimiento de la versión secuencial y paralela de un algoritmo genético, variando tanto el lenguaje de programación como la cantidad de núcleos utilizados. Para ello, se seguirá una metodología rigurosa que permita validar la funcionalidad y medir de manera precisa el rendimiento en diversas configuraciones.

Variables del Experimento:

- **Variable dependiente:** El tiempo de ejecución del algoritmo genético (en segundos), que será la principal métrica de rendimiento evaluada.
- **Variables independientes:**
 - **Lenguaje de programación:** Se seleccionarán varios lenguajes de programación que soporten tanto versiones secuenciales como paralelas del algoritmo (por ejemplo, Python, C++, Java).
 - **Cantidad de núcleos:** Se variará el número de núcleos de procesamiento en la máquina entre 1 (para la versión secuencial), 2, 4, 8, y más, dependiendo de las capacidades del sistema.

Metodología de la Ejecución:

- Por cada combinación de lenguaje de programación y cantidad de núcleos, se ejecutará el algoritmo en la misma máquina, garantizando así que los resultados no estén afectados por diferencias en el hardware. Esto asegura consistencia en las condiciones experimentales.
- Para cada configuración, se tomarán tres o cinco mediciones del tiempo de ejecución, con el fin de obtener una media estadística representativa del rendimiento. De esta forma, se reduce la influencia de cualquier fluctuación ocasional en los tiempos de ejecución.

Validación de la Funcionalidad:

- La funcionalidad del algoritmo genético se validará asegurando que todas las versiones, tanto secuenciales como paralelas, converjan a una solución válida o cercana al óptimo para el problema planteado. Se verificará que las soluciones finales sean consistentes entre versiones, lo que garantizará que la paralelización no altere el comportamiento del algoritmo, sino que únicamente afecte a su velocidad de ejecución.
- Se realizará una comparación cualitativa de las soluciones obtenidas en cada versión, evaluando la calidad y precisión para garantizar que la paralelización no afecte negativamente el proceso evolutivo del algoritmo.

Validación del Rendimiento:

- Se utilizarán métricas clave para validar el rendimiento de la solución paralela en comparación con la secuencial. Estas métricas incluyen:
 - Speedup: Relación entre el tiempo de ejecución secuencial y el tiempo de ejecución paralelo. Esta métrica indicará cuánta mejora en velocidad se ha obtenido al agregar más núcleos.
 - Eficiencia: Relación entre el speedup y el número de núcleos. Indicará qué tan bien se está aprovechando el aumento en la cantidad de núcleos.
- El uso de métodos estadísticos permitirá validar si las diferencias en rendimiento entre versiones son estadísticamente significativas. Se aplicarán pruebas de significancia (como pruebas t de Student o análisis ANOVA) para determinar si el rendimiento de la versión paralela ofrece mejoras sustanciales en comparación con la secuencial.

Condiciones Experimentales Controladas:

- El experimento se realizará en condiciones controladas, con todas las pruebas ejecutadas en una misma máquina con especificaciones definidas (procesador, RAM, sistema operativo) para evitar variaciones en el hardware que puedan afectar los resultados.
- Se tomarán medidas para minimizar interferencias externas que puedan alterar los tiempos de ejecución, como la ejecución de otros procesos en segundo plano durante las mediciones.

Análisis de Resultados:

- Los resultados obtenidos se analizarán usando gráficos comparativos y estadísticas descriptivas, lo que permitirá visualizar el impacto de las diferentes configuraciones de paralelización.
- Además, se incluirá un análisis detallado de cuándo y bajo qué condiciones la versión paralela ofrece beneficios significativos sobre la secuencial, y se identificarán las limitaciones que puedan surgir cuando el número de núcleos incrementa sin una mejora proporcional en el rendimiento.

Cronograma

	OCTUBRE				NOVIEMBRE			
ACTIVIDAD	SEM 1	SEM 2	SEM 3	SEM 4	SEM 1	SEM 2	SEM 3	SEM 4
Propuesta Proyecto								
Consultas algoritmos y lenguajes								
Implementación Algoritmo								
Resultados y análisis								
Presentación								

Referencias

1. "Algoritmo genético". MathWorks - Creador de MATLAB y Simulink - MATLAB y Simulink - MATLAB & Simulink. Accedido el 16 de octubre de 2024. [En línea]. Disponible: <https://es.mathworks.com/discovery/genetic-algorithm.html>
2. La universidad en internet. "La computación paralela: características, tipos y usos". UNIR. Accedido el 16 de octubre de 2024. [En línea]. Disponible: <https://www.unir.net/revista/ingenieria/computacion-paralela/>