

Proyecto Final

Curso de Sistemas Operativos y Laboratorio

Técnicas de aprendizaje automático para
sistemas operativos

Diego Alejandro Castañeda Ossa

Jose Carlos Ortiz Padilla

Resumen

En el contexto actual de la informática, los sistemas operativos enfrentan desafíos significativos relacionados con la gestión eficiente de recursos y el scheduling de procesos. Este proyecto propone explorar e implementar técnicas de aprendizaje automático (machine learning, ML) para optimizar estas tareas fundamentales en un sistema operativo.

Introducción

En la actualidad, los sistemas operativos enfrentan una creciente demanda de eficiencia debido al aumento en la complejidad de las aplicaciones y la diversidad de entornos de ejecución, como la computación en la nube, el Internet de las Cosas (IoT) y los dispositivos móviles. La gestión óptima de recursos y el scheduling de procesos son desafíos críticos que requieren soluciones innovadoras. El uso de técnicas de aprendizaje automático puede revolucionar la forma en que los sistemas operativos abordan estos problemas, permitiendo una adaptación dinámica y proactiva a las condiciones cambiantes del sistema. Así, el desarrollo de este desafío se vuelve esencial en el contexto tecnológico actual, donde la agilidad es prioritaria para el éxito de las plataformas informáticas.

Antecedentes o marco teórico

Es fundamental considerar varios aspectos teóricos, entre los cuales destacan los conceptos de **scheduling de procesos**, **gestión de memoria** y **gestión de recursos**. El scheduling de procesos se refiere a la forma en que un sistema operativo asigna tiempo de CPU a los diferentes procesos en ejecución, y es crucial para optimizar el uso del procesador y garantizar una respuesta rápida a las solicitudes de los usuarios. La gestión de memoria, por su parte, implica la asignación eficiente de espacio de memoria a los procesos y el manejo de la memoria virtual, lo que puede beneficiarse del aprendizaje automático al prever patrones de uso y optimizar la carga y descarga de datos. Además, la gestión de recursos es esencial para asegurar que múltiples hilos y procesos accedan a los recursos compartidos de manera eficiente y sin conflictos. La teoría del aprendizaje automático proporciona herramientas y algoritmos que pueden ser aplicados en estos contextos, permitiendo a los sistemas operativos adaptarse y mejorar continuamente su desempeño en base a datos históricos y patrones de uso.

Esta relación con los temas del curso de Sistemas Operativos se manifiesta en la necesidad de entender cómo funcionan internamente los sistemas operativos y cómo pueden integrarse algoritmos de aprendizaje automático para mejorar su funcionamiento. Por ejemplo, al aplicar técnicas de aprendizaje automático en el scheduling de procesos, se puede crear un sistema que no solo responda a reglas predefinidas, sino que también aprenda de la carga de trabajo y

los tiempos de respuesta, ajustando dinámicamente las prioridades de los procesos en ejecución. Esto implica una comprensión profunda de cómo se gestionan los procesos y la memoria, así como de la estructura y el comportamiento de los sistemas operativos en general.

Objetivos (principal y específicos)

Desarrollar e implementar un sistema que integre técnicas de aprendizaje automático en el scheduling de procesos de un sistema operativo, con el fin de optimizar la asignación de recursos y mejorar el rendimiento general del sistema.

- **Investigación y análisis de algoritmos de Scheduling:** Investigar los diferentes algoritmos de scheduling existentes y analizar sus ventajas y desventajas, para identificar oportunidades de mejora mediante el uso de técnicas de aprendizaje automático.
- **Desarrollo de un modelo predictivo:** Crear un modelo predictivo utilizando algoritmos de aprendizaje automático que sea capaz de anticipar la carga de trabajo y el comportamiento de los procesos, permitiendo ajustar dinámicamente las prioridades en el scheduling.
- **Implementación del modelo:** Implementar el modelo predictivo y probar su funcionalidad utilizando ejemplos de ejercicios de scheduling desarrollados en Python, simulando la asignación de procesos y evaluando su comportamiento en diferentes escenarios.
- **Evaluación del rendimiento:** Realizar pruebas comparativas entre el sistema de scheduling tradicional y el sistema mejorado con aprendizaje automático, evaluando métricas como el tiempo de respuesta, la utilización de CPU y la eficiencia en la gestión de recursos.
- **Documentación y análisis de resultados:** Documentar el proceso de implementación y los resultados obtenidos, analizando el impacto del aprendizaje automático en el rendimiento del modelo y proporcionando recomendaciones para futuras mejoras.

Metodología

Para la implementación del proyecto, las siguientes herramientas serán clave:

- **Lenguajes de programación:** Python para desarrollar el modelo predictivo y su implementación con ejemplos.

- **Frameworks de machine learning:** TensorFlow o Scikit-learn para desarrollar y entrenar el modelo de aprendizaje automático.
- **Git/GitHub:** Para la gestión de versiones del código y colaboración en el desarrollo.
- **Entornos de desarrollo:** VS Code para la programación y pruebas del sistema.

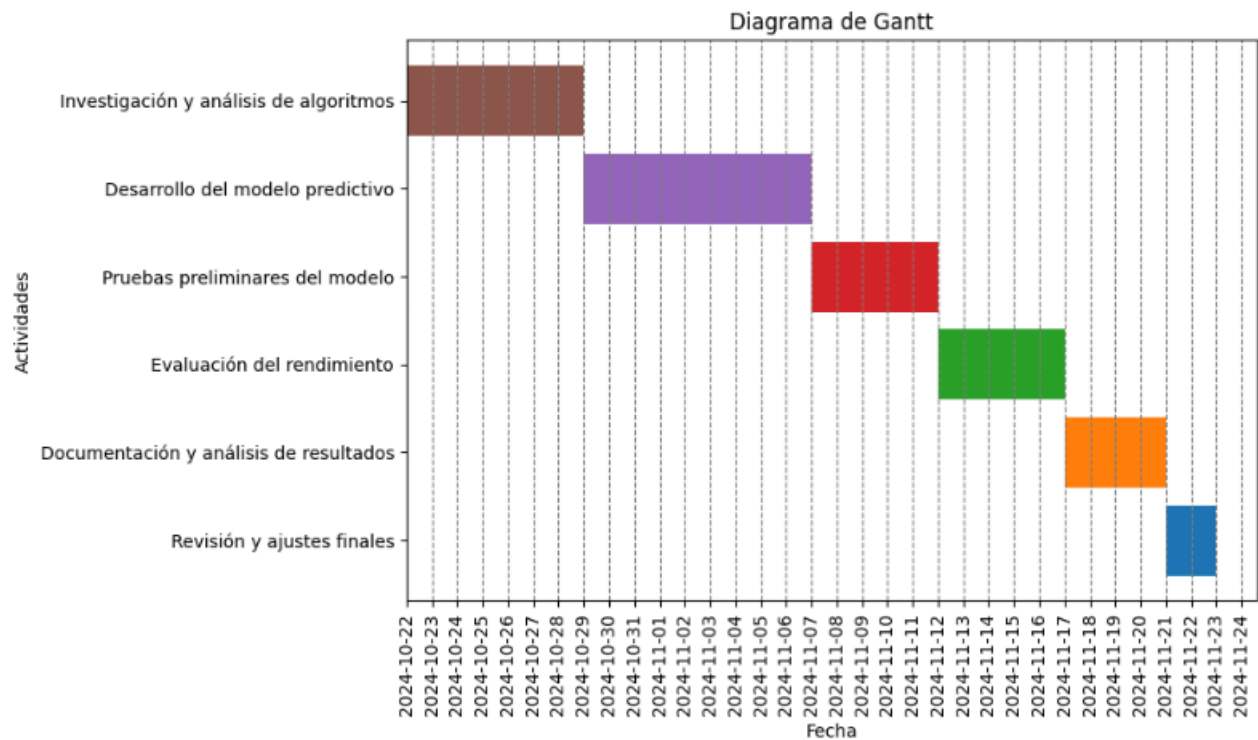
El desarrollo del proyecto seguirá un flujo lógico de actividades, asegurando que cada paso cumpla con los objetivos propuestos:

- **Investigación y análisis:** Investigar los algoritmos de scheduling de sistemas operativos (FIFO, Round Robin, etc.) y analizar sus características para identificar oportunidades de mejora con machine learning.
- **Desarrollo del modelo predictivo:** Crear un modelo de aprendizaje automático que pueda predecir la carga de trabajo y optimizar la gestión de los procesos. Entrenar el modelo utilizando datos simulados o datos históricos de procesos.
- **Pruebas del modelo:** Evaluar el rendimiento del modelo de manera independiente para asegurar que hace predicciones precisas.
- **Evaluación del rendimiento:** Comparar el rendimiento del sistema mejorado con el modelo predictivo y el sistema sin mejoras, utilizando métricas como uso de CPU, tiempo de respuesta, y eficiencia general.
- **Documentación y análisis de resultados:** Documentar todo el proceso de desarrollo, integración, y pruebas, junto con los resultados obtenidos para el análisis final del proyecto.

Diseño de Experimentos

Para validar la funcionalidad y rendimiento de la solución en este desafío, se puede diseñar un experimento que se centre en la comparación entre el sistema de scheduling tradicional del sistema operativo y la versión mejorada con aprendizaje automático. Es importante realizar pruebas rigurosas bajo distintas condiciones de carga de trabajo, midiendo métricas clave como el tiempo de respuesta, la utilización de CPU y la eficiencia general.

Cronograma



Referencias

Akgun, I. U. (2022). *Using machine learning to improve operating systems' I/O subsystems* (Technical Report FSL-22-02). Stony Brook University.
<https://www.fsl.cs.stonybrook.edu/docs/kml/umit-prelim-tr-fsl-22-02.pdf>