Introduction

In the realm of systems programming, C and Rust represent two significant paradigms of low-level language design. While C has been a cornerstone of systems programming for decades, Rust emerges as a modern alternative promising memory safety without sacrificing performance.

Recently, Rust is often noted for its strong type system, memory safety guarantees, and modern concurrency features, yet it's generally thought to be slightly slower than C. This paper aims to provide a comprehensive comparative analysis of their performance differences. Understanding the performance slowdowns is particularly relevant to modern operating systems; Linux and Windows have been integrating Rust code into their kernels, several new operating systems in Rust are quickly gaining popularity, and world governments are pressuring companies to transition to memory-safe languages in the next few years. Therefore, having a good understanding of the performance challenges that Rust is facing, and especially their causes, is a relevant and important issue.