


C++ and Safety

Timur Doumler

 @timur_audio

CppOnSea
29 June 2023



*An artist's conception of a supernova explosion.
Credit: NASA's Goddard Space Flight Center / ESA / Hubble / L. Calçada*



Timur Doumler

Borrowing Trouble: The Difficulties Of A C++ Borrow-Checker



Authors: danakj@chromium.org, lukasza@chromium.org, palmer@chromium.org
Publication Date: 10th September 2021

Introduction

A common question raised when comparing C++ and Rust is whether the Rust borrow checker is really unique to Rust, or if it can be implemented in C++ too. C++ is a very flexible language, so it seems like it should be possible. In this article we'll explore if it is possible to do borrow checking at compile time in C++.

Some background on C++ efforts

Many folks are working on [improving C++](#), including improving its memory safety. [Clang](#) has [experimental -Wlifetime warnings](#) to help catch a class of use-after-free bugs. The cases it catches are typically [dangling references to temporaries](#), which makes them a valuable set of warnings to enable when it is available. But the cases it would solve do not seem to intersect with the set of cases [MiraclePtr](#) is attempting to protect against, which is an effort to frustrate