Color coating Rust – protecting the exposed parts

How can Memory Tagging fill the holes left by the Rust compiler?

The setting:
• Rust offers a memory safe programming environment except for the unsafe “escape hatch”
• Arm Memory Tagging promises to catch most memory errors by coloring unsafe allocations

The problem:
• No coloring support in Rust compiler
• Existing LLVM safety analysis does not recognize Rust safety guarantees, decreasing performance and protection for some safe parts in the program.

Our solution: Rust safety analyzer and memtag sanitizer support
• Augment existing safety analysis with Rust level analysis
• 60% reduction in stack allocations that need to be colored

Joint work with Huawei

In progress