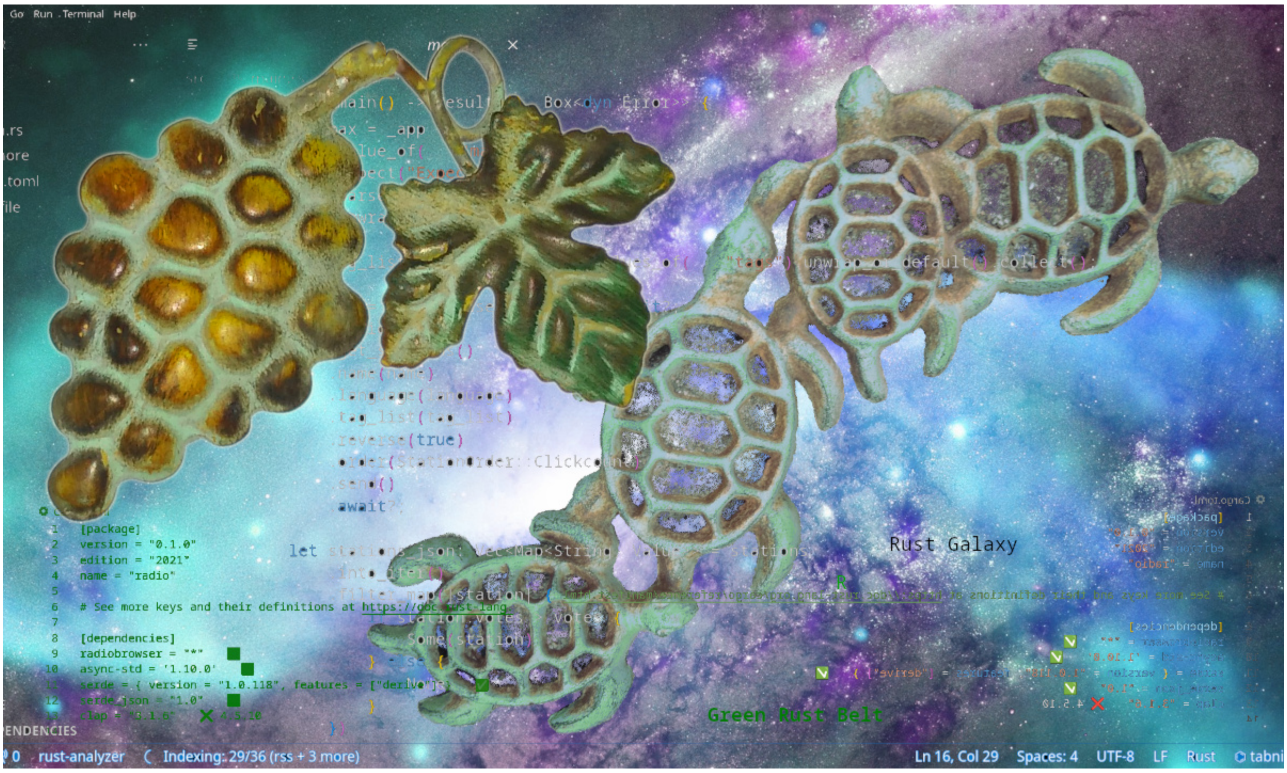
Green Rust

How the Rust Programming Language Saves Energy

August 09 2024



How does the Rust programming language save energy? We answer that question here and now.

What is a Rust?

Rust is a general purpose programming language released in 2015. Rust's key traits are:

- Performance: Ultra fast rivaling C
- Memory Efficient: Rust is extremely memory efficient, and has no garbage collector
- Reliable: Rust is memory and thread safe
- Works in computers and embedded devices
- Productive: Great docs, compiler messaging, package manager, editor support

We'll expound on these ideas next.

Rust Security Saves Energy!

Rust has good security features built into it that make it superbly resilient to cybercrime. Better security induces better code efficiencies.

Memory Security

Rust has a keen sense of memory security. To understand why this is important, we must note that over 70% of all bugs in recent times are memory issues. These are issues that Rust overcomes by imposing strict guardrails in the compiler. Recall that a compiled language (i.g., Rust) gets transformed by the compiler into machine code that the computer knows how to run.

This memory safety implies:

- Memory overflows are not easily be exploited as in C and C++
- Software will use a minimal amount memory
- OS crashes in Rust will be rare
- Memory leaks, that cause a buildup of memory use, are virtually eliminated

These features reduce downtime and security breaches responsible for poor energy efficiencies.

Type Safety

Rust is statically typed, meaning variable types are well known, static, and stable. This prevents type mismatches and unexpected behavior that cause security issues.

Borrow Checker

Rust has a strict data ownership model, that ensures data integrity. This leads to more stable code that avoids downtime or race conditions, and saves energy in the long term.

Rust Performance Saves Energy!

Rust is aggressively performance conscious. Well performing code runs faster with minimal energy. Here is how:

Reduced CPU Usage

Rust generates optimized code through a variety of means:

- Efficiently Optimized: The compiler takes advantage of optimizations via various means, and makes the code lean and mean.
- Zero Cost Abstraction: Developers write the code naturally and readably. The compiler figures out how to optimize the code as if it was written with performance in mind. This reduces run time, but also developer energy expense.
- Smaller Memory Footprint: Overall code is smaller in memory, minimizing time and energy to load and run.
- Scalability: Rust excels in multi-thread and multi-core systems making it more efficient on distributed systems.

Rust Embedded Goodness

Embedded systems are the backbone of renewable and conventional energy systems. Power plants ops, storage, grids, turbines, distribution, and safety systems all rely on embedded controllers.

Rust is embedded ready: https://www.rust-lang.org/what/embedded. There are already many embedded libraries and platforms for Rust (https://github.com/rust-embedded/awesome-embedded-rust). Add to that a large and growing groups of Rust embedded engineers (https://blog.rust-lang.org/2024/02/19/2023-Rust-Annual-Survey-2023-results.html).

Fundamental reasons Rust is ideal for embedded:

- Performance
- Interoperable with C
- Concurrency Safety
- Memory Flexibility and Safety
- Powerful Static Analysis
- Productivity: rich standard library, excellent build system (Cargo)

Linux is Getting Rusty

Linux developers now realize that Rust is the future for software security, performance, and development. This means that new code will be written in Rust, and that older code will be written in Rust.

- https://www.theregister.com/2022/10/05/rust_kernel_pull_request_pulled/

This implies a more efficient and more secure ecosystems, because upwards of 90% of the cloud is Linux based. As the Linux transformation will take decades, just a fraction will be written/rewritten in Rust over the next 10 years.

FYI, there already exist operating systems written (partially or fully) in Rust:

- https://github.com/flosse/rust-os-comparison

Rust has Graphical User Interfaces

Rust GUI apps are ideal to optimize energy used by huge margins (50% to 95%), over otherwise web-based or Java applications. Not all apps will be this efficient but the potential is there for energy savings.

Remember our goal is to save energy globally, at the server, desktop, and mobile. Think of the millions of computers running inefficient apps in browsers.

Final Thoughts

The world is increasingly reliant on compute power, and is punished for inefficient code and security breaches. Rust is a secure and energy efficient language that increases security and performance, thus saving tons of CO2 and energy.

Some final links to ponder:

- https://www.theregister.com/2024/02/08/rust_software_memory_safety/
- https://iotac.eu/energy-consumption-by-ict-and-cybersecurity-at-the-time-of-cop26/
- https://www.theregister.com/2021/11/30/aws_reinvent_rust/
- https://aws.amazon.com/blogs/opensource/sustainability-with-rust/
- https://www.informationweek.com/software-services/how-to-shrink-it-s-carbon-footprint-using-software

If you have questions or comments you can reach us at https://fortuitous.com