

Increasing Language Performance through Template Runtime Libraries

Friday 19 September 2025 11:05 (13 minutes)

Programming languages employ significant diversity in design philosophies and implementation of their runtimes.

Languages such as Java and CPython utilize distinct virtual machine and interpreter environments that enable high-level features but require substantial runtime overhead, while others like Go, Rust, C++, and C provide offer system-level programming and are compiled to binary files, requiring smaller runtimes.

This creates a gap between high-level constructs (such as garbage collection and built-in data structures) and low-level functionality, with memory safety being present at both ends of the spectrum, with Python and Java at one end and Rust and Go at the other.

We propose a novel approach to designing runtimes in languages that support generic programming by transforming the D programming language's runtime library from a monolithic architecture to a template-based implementation by converting its runtime hooks from TypeInfo-based implementations to templates.

We evaluated the performance of each hook with the template and non-template implementations and increased the running performance of applications by as much as 25% at the cost of a 10-30% increase in compilation time.

This work may also extend the language features available when compiling for IoT devices, which, due to their low available memory, do not support the full runtime of the D language.

Author: DUȚU, Teodor-Ștefan (National University of Science and Technology Politehnica Bucharest)

Co-authors: NITU, Razvan (University POLITEHNICA of Bucharest); STANILOIU, Constantin Eduard (University POLITEHNICA of Bucharest); Mr ALEXANDRESCU, Andrei (Nvidia Corporation)

Presenter: DUȚU, Teodor-Ștefan (National University of Science and Technology Politehnica Bucharest)

Session Classification: Doctoral Symposium

Track Classification: Security & Resilience in Cyber-Physical Systems