

STUDENTŲ ĮTRAUKIMO Į MOKSLINĘ VEIKLĄ SKATININAMOJO KONKURSO TEMA

Temos pavadinimas: Lygiagretaus programavimo naudojant Rust programavimo kalbą efektyvumo ir saugumo tyrimas
Tikslas: Empiriškai įvertinti Rust programavimo kalbos efektyvumą lygiagrečiame programavime
Trumpas temos vykdymo aprašymas (ne daugiau kaip 2000 ženklų): Lygiagretusis programavimas yra būtinas norint pilnai išnaudoti šiuolaikines daugia-branduolines procesorių (angl. multicore CPU) architektūras. Tačiau jis kelia didelių iššūkių dėl duomenų lenktynių (angl. data races), sinchronizacijos sąnaudų ir atminties saugumo problemų. Šis projektas siekia įvertinti programavimo kalbos Rust tinkamumą lygiagrečiam programavimui. Rust savybės žada „bebaimį lygiagretumą“ (angl. fearless concurrency), potencialiai gali užtikrinti tiek aukštą našumą, tiek stiprias saugumo garantijas be šiukšlių surinkimo mechanizmo (angl. garbage collection). Tyrimo metu numatoma sukurti testinių (angl. benchmark) programų rinkinį (pvz., matricų daugybos, grafų algoritmų, duomenų apdorojimo užduočių), realizuotas naudojant Rust ir C++ kalbas. Kiekviena realizacija bus analizuojama pagal vykdymo našumą, atminties naudojimo efektyvumą, gijų saugumą ir kodo sudėtingumą. Taip pat bus vertinami programavimo patirties rodikliai – kodo skaitomumas, derinimo sudėtingumas ir klaidų aptikimo galimybės. Tikimasi, kad šio darbo rezultatas bus išsamiai Rust programavimo kalbos tinkamumo lygiagrečiam programavimui analizė akademiniame ir pramoniniame kontekste.
Temą siūlantis mokslininkas/dėstytojas: Vadimas Starikovičius (FMF Matematinio modeliavimo katedra, Lygiagrečiųjų skaičiavimų laboratorija)

TOPIC OF A COMPETITION PROMOTING STUDENT ENGAGEMENT IN SCIENTIFIC ACTIVITIES

Topic: Exploring the Efficiency and Safety of Parallel Programming in Rust
Goal: The goal of this project is to empirically evaluate the Rust programming language's effectiveness for parallel programming.
Short description (max. 2000 characters): Parallel programming is essential for leveraging modern multicore architectures, yet it remains challenging due to data races, synchronization overhead, and memory safety issues. This project aims to evaluate the effectiveness of Rust as a language for parallel and concurrent computing. Rust's ownership and borrowing system promises "fearless concurrency," potentially offering both high performance and strong safety guarantees without garbage collection. The research will involve designing and implementing a set of benchmark applications (e.g., matrix multiplication, graph algorithms, and data processing tasks) using Rust and C++. Each implementation will be analyzed in terms of execution performance, memory efficiency, thread safety, and code complexity. Additionally, developer experience metrics such as code readability, debugging difficulty, and compile-time error handling will be assessed. The expected outcome is a comprehensive evaluation of Rust's suitability for parallel programming in academic and industrial contexts.
Supervisor researcher/lecturer: Vadimas Starikovičius (Department of Mathematical modelling, Laboratory of Parallel Computing)