
Annotations for Safe Parallelism

Alexandros Tzannes, Michael Han, Vikram S. Adve

Goal

- Checker guarantees that parallelism in C++ code is safe
- Use this checker on real code (>1M LOC)
 - Scalability
 - Annotation Burden

Basics

- Based on DPJ (Deterministic Parallel Java)
 - Region and Effect annotations (attributes)
 - Strong Guarantees (race freedom, ...)
- Fork-Join Parallelism
 - Scoped Locks & Atomic Sections
- C++ challenges
 - Pointer Arithmetic & Aliasing, Implicit Functions, ...
- Library Support
 - Context annotations

Plan

1. Basic Checker

- Deal with C++

2. Minimize Annotation Burden

- Smart defaults for annotations
- Infer effect & region annotations (interprocedural)

Interactive Parallelization

- Programmer:
 1. creates parallel tasks
 2. asks checker if it is safe
- Checker replies:
 1. YES, or
 2. NO, because X & Y may cause race, or
 3. Cannot prove safety because XYZ
 - programmer fixes this by adding annotation