Robustness Between Weak Memory Models

Soham Chakraborty

Presentation by
Akshay Gopalakrishnan

September 12, 2021

Introduction

- Robustness is a property that a program can have given two memory models.
- A program executing under a memory model K is Robust against memory model M if all consistent executions under K are also consistent under M.
- This property is useful in ensuring safe portability of programs across different platforms.
- However, as far as memory models are concerned, Robustness is maintained only against Sequential Consistency (SC).
- This paper analyzes robustness against weaker memory models such as x86-TSO, ARMv7 and ARMv8.
- Robustness conditions are established against each of these models.
- An analysis tool is also developed to ensure a program is Robust or if not, adding fence instructions to ensure Robustness.

Examples of Robustness

Robustnes Definition

A program P is M-K robust if all its K-consistent executions are also M-consistent.

Axiomatic Specification of Memory Consistency Models

- Define binary relations (mainly partial orders) between program constructs Read/Write/RMW/Fence.
- Per-execution semantics.
- Specify axioms based on relations, which are acyclic/irreflexivity conditions on execution graphs.

Preliminary Definitions of Binary Relations

SC and X86-TSO

Robustness Conditions: Intuition

SC-x86TSO Robustness Condition

Checking and Enforcing Robustness

- Take the Control Flow Graph of the program.
- Build a Memory-access pair graph (MPG) capturing two important edge relations (eco and epo).
- If MPG has a cycle, check each access pair whether ordered, as per a condition based on M-K Robustness.
- If every pair is ordered, program is M-K Robust.
- Else, insert appropriate fences between memory access pairs that are not ordered.

Experimental Evaluation

Thank you

Questions?