Static Analysis Based Fence Synthesis

CS4560 - Parallel and Concurrent Programming

February, 2026

Project description

In this project, you will synthesize fence instructions to enforce TSO and PSO consistency models.

TSO consistency model forbids reordering of write-write, read-write, and read-read access pairs. Note that it allows write-read access pairs.

PSO consistency model forbids reordering of read-write, and read-read access pairs. Note that it allows write-write and write-read access pairs.

[1]

Background Information

C11 concurrency

C/C++ defines a relaxed memory concurrency model known as the C11 concurrency model [1]. C11 has various kinds of accesses that affect shared memory concurrency. To begin with, it provides plain or non-atomic load and store accesses. In addition, C11 also has atomic accesses of four kinds: load, store, atomic update (RMW) such as compare-and-swap and atomic increment, and memory fence. Each atomic access is attached with a memory order from – relaxed, acquire, release, acquire-release, and sequentially-consistent.

However, in this project the memory orders are irrelevant.

LLVM

LLVM compiler also introduces the same accesses to compile C11 in its intermediate representation (IR). It is possible to analyze and transform LLVM IR by inserting new instructions.

Note: *feel free to use any other tool if you like.*

Projects

In this project, you will synthesize and insert SC-fence instructions in LLVM IR by analyzing the programs with control flow to enforce TSO and PSO properties. We want to insert an optimal number of fences to avoid any redundant fences as it is a costly operation.

Roadmap for the project:

The project involves the following steps:

- Set up the LLVM compiler. It is available at `https://github.com/llvm/llvm-project`.
- Understand the internals of the LLVM compiler.
- Develop the algorithms for fence synthesis.
- Implement the algorithm inside LLVM.
- Evaluate on the LLVM benchmarks*.

You may consider other compilers as well.

[1] The project description is subject to small changes and updates. Please get in touch with the TAs and the teachers if you have any questions.

Note

- Build LLVM with clang/clang++ for compiling C/C++ programs.

Restriction Do not change the memory orders.

References

[1] Mark Batty, Scott Owens, Susmit Sarkar, Peter Sewell, and Tjark Weber. Mathematizing C++ concurrency. In *POPL'11*, pages 55–66. ACM, 2011.