# Hydra LLVM: Instruction Selection with Threads

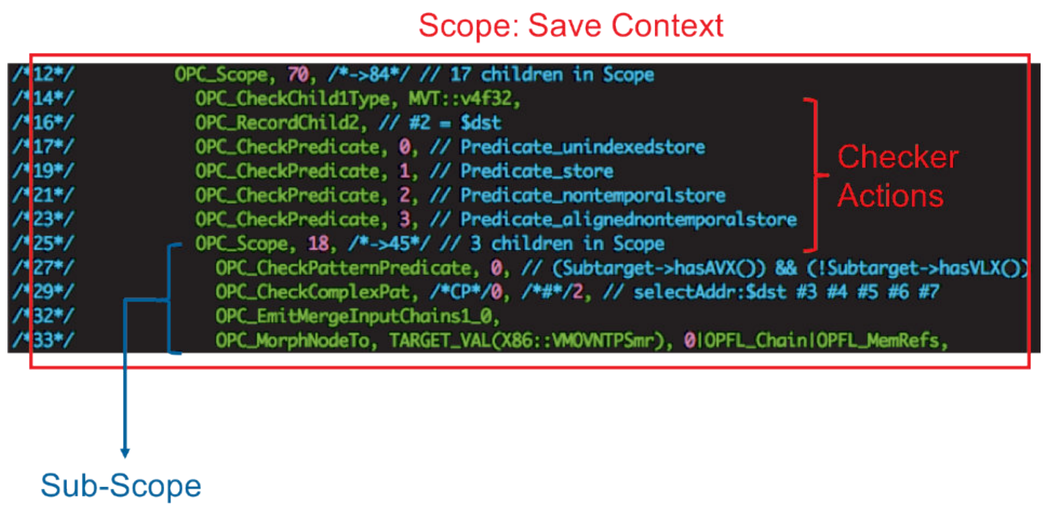
Min-Yih Hsu and Prof. Jenq-Kuen Lee,
Department of Computer Science, National Tsing-Hua University, Taiwan
{myhsu, jklee}@pllab.cs.nthu.edu.tw

## Instruction Selection is SLOW!

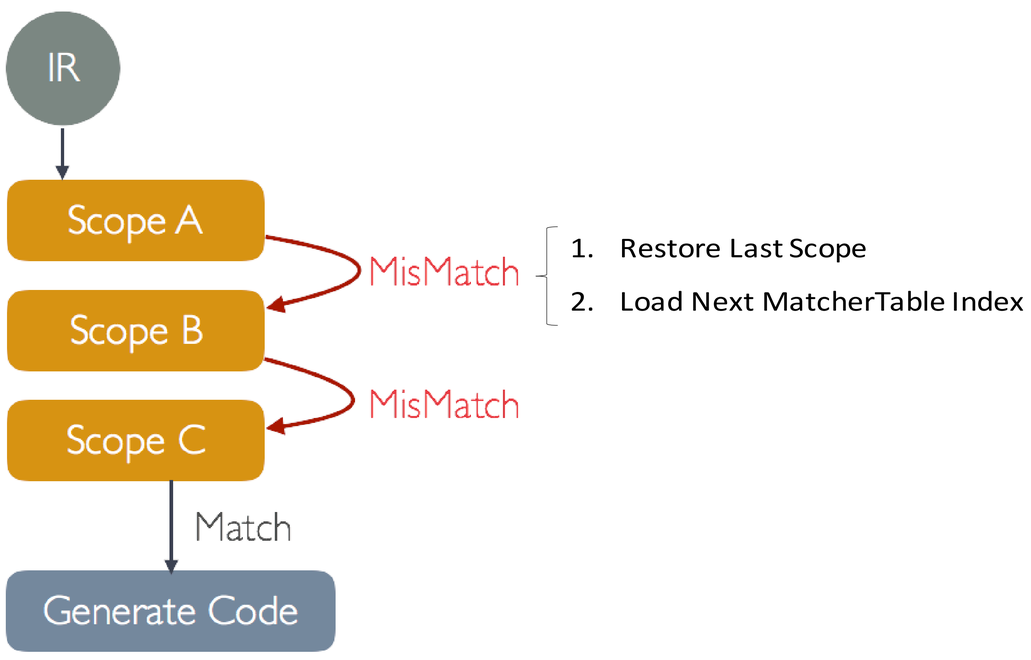
By the rise of program complexity and some specific usages like JIT(Just-In-Time) compilation, compilation speed becomes more and more important in recent years. Instruction selection in LLVM, on the other hand, is the most time-consuming part among all the LLVM components, which can take nearly 50% of total compilation time

## Instruction Selector in Current LLVM

LLVM has a greedy based instruction selector, which is basically a bytecode interpreter

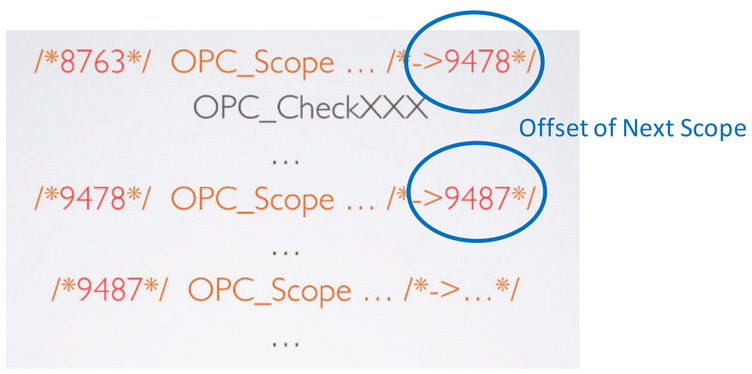



In the interpreter, it uses checkers to perfrom pattern matching, and use scopes to provide a local context for a subset of checkers. Each scope would jump to next scope if a failure is raised within it.

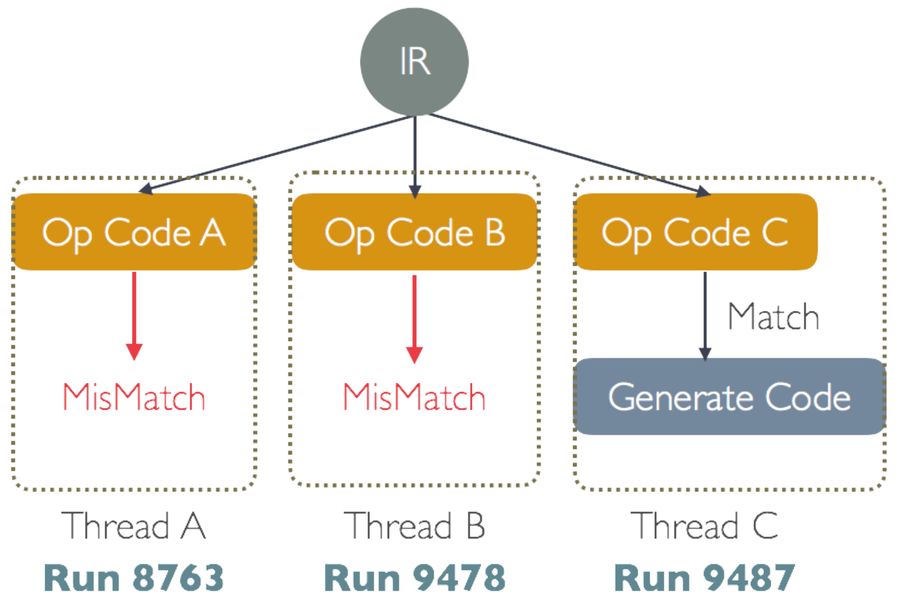



## Parallelizing Selection Scopes with Threads

Each selection scope has the offset to the next scope, which is statically known ahead-of-time, thus provide us the basis of parallelizing the instruction selection process efficiently.




With the statically-known scope offset, we adopt a speculation based approach that execute every possible selection scopes simultaneously in threads




## Experiment Results