

LLVM & HPSSA

Hot Path SSA Form in LLVM

Presented By Abhay¹ & Muzzammil¹

¹IIT Kanpur
PRAISE Group

Dr. Subhajit Roy, Dr. Awanish Pandey, Mr. Sumit Lahiri

What we modified in LLVM?

- New `llvm::intrinsic` signature, `"llvm.tau"`.
- Modified `Verifier::verifyDominatesUse()` since we don't want our intrinsic to interfere with `dominators` computation.

What we modified in LLVM?

- New `llvm::intrinsic` signature, `"llvm.tau"`.
- Modified `Verifier::verifyDominatesUse()` since we don't want our intrinsic to interfere with `dominators` computation.

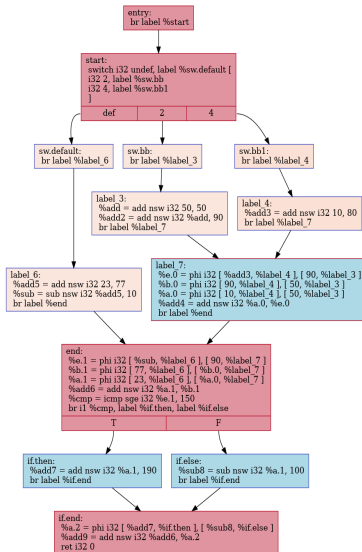
HPSSAPass : Overview

- New `llvm::HPSSAPass` pass using the new Pass Manager.
- Pass runs over a `llvm::Function` and inserts "`llvm.tau`" intrinsic calls with speculative and safe arguments.

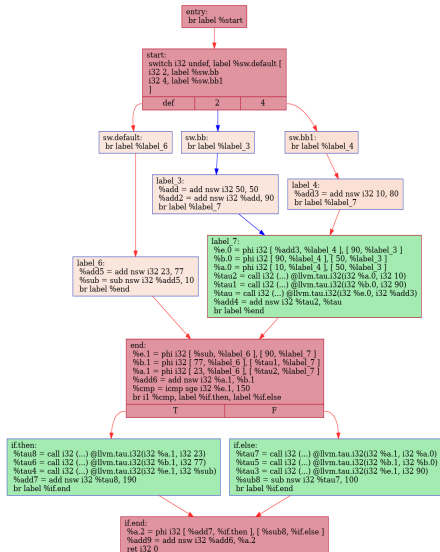
Key HPSSA Data Structures :

- Hot Path Set using `llvm::BitVector`.
- Definition Accumalator, `defAccumalate` as a map
`std::map<{PHINode*, BasicBlock*}, {Value*, BitVector}>`.
- Variable Renaming Stack as a map,
`std::map<Value*, Value*>`

HPSSA Transformation



CFG for 'main' function before running HPSSAPass (no tau)



CFG for 'main' function after running HPSSAPass

HPSSAPass : Auxilliary Functions

- HPSSAPass::getProfileInfo(Function \&F)
- HPSSAPass::getCaloricConnector(Function \&F)
- HPSSAPass::Search(BasicBlock \&BB, DomTreeNode \&DTN)

HPSSAPass : Main & Destruction Pass

- `HPSSAPass::run(Function &F, FunctionAnalysisManager &AM)`
- `llvm::Function::RPOT()`.
- `llvm::successors()`.
- `llvm::DominatorTreeAnalysis` and `llvm::dominates()`.
- Replace use of `phi`'s with `tau` variables using renaming stack.
- Out of HPSSA Form.

SSCCP Pass

LLVM Changes

SSCCP with an Example