

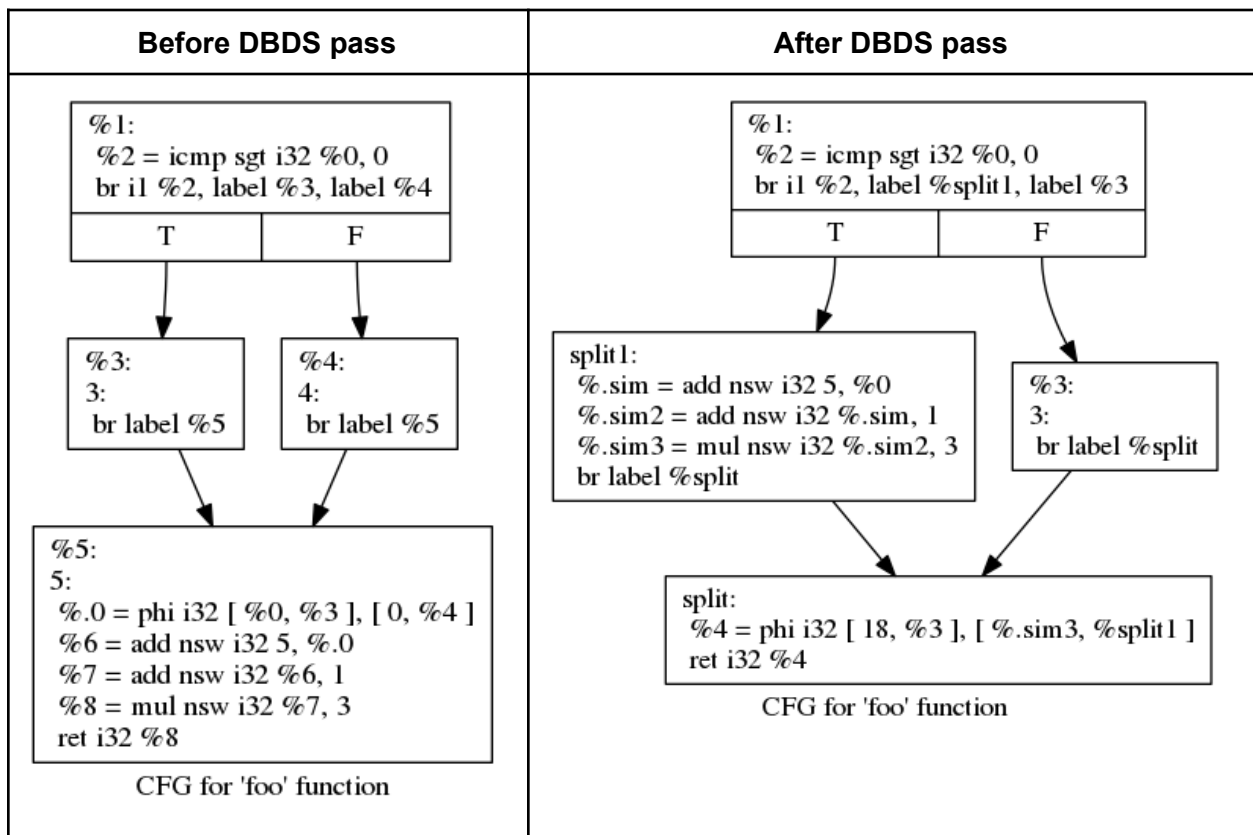
Major Changes

The only major change that we made was to add “apply the resulting optimization” to our list of tasks as part of the 75% goal. It is an important part of the process, since otherwise our simulations cannot actually affect the output code.

The only other change is that we’re using LLVM’s built-in constant folding engine so we correctly support the various arithmetic operations instead of hand-coding a small, buggy subset of them. As such, it was trivial to support both addition and other operations.

What we have accomplished so far

We are exactly on the (revised) schedule: we are done with the DBDS simulation, constant folding, and applying the simulation results to the generated code. See below for an example of code duplication enabling significant constant folding:



Meeting our milestone

We met our milestone, as I said above. We have a working simulation framework, simulated constant folding, and application of our simulated results on the final code output.

Surprises

We were surprised by how smoothly the simulation process went. Iterating over the basic blocks, even in Dominance Tree order, was pretty easy. Also, it was not difficult to use LLVM's constant folding function with a new instruction that was generated for the simulation.

On the other hand, applying the results of the simulation was shockingly difficult. It's extremely fiddly to rearrange the CFG such that a basic block is bypassed. For example, variables that were generated in the duplicated block could be accessed anywhere later in the CFG, necessitating creation of a "landing pad" block that has phi nodes for these variables. Then we need to replace only the correct uses of that variable with the phi function (only phi nodes in the duplicated block, non-phi nodes in the "landing pad" block, and all other external uses).

Revised schedule

You can see the revised checklist on our website, but it and our schedule are very much on track. We foresee no problem meeting, or more likely beating, our schedule for implementing more optimizations.

Resources needed

We have all the resources we need to be successful.