Write-and-Do-Mini-Assignment#3: MLIR and LLVM

Sagar Jain
CS17BTECH11034

September 14, 2019

MLIR in LLVM

As mentioned in here, recently there was a massive announcement to include **MLIR** in **LLVM** as a subproject. Having MLIR in LLVM makes sense to me for the following reasons:

1. MLIR uses LLVM IR as one of its primary code generation targets already.
2. One of the main draw-backs of LLVM today is considered to be the loss of information in its IR, including MLIR in the LLVM project solves this problem by providing an infrastructure to build these pre-LLVM implementations.
3. MLIR has advance analysis(example, loops) capabilities, these moved into the pre-existing llvm analysis will make it usable both by the MLIR-level transformations along with providing llvm access to these results to use in its own transforamtions later on.
4. LLVM will also benefit greatly from this inclusion, it would now have be at the forefront whether it be heterogenous computing or accelerators, MLIR can facilitate it in many ways.

I would also like to express my concern about the project structure and developemnt. I hope care would be taken to not bloat the pre-llvm transformations which would adversely impact not only the running time of the compiler but also the development time since the project is already a vast one. Documentation would also be have to taken care of since a lot of MLIR classes are just re-exported from LLVM. Finally, although already mentioned by Chris in the discussion that the sub-project would follow LLVM Developer Policy, a lot of contributors moving in from MLIR hopefully doesn't alter the LLVM project intrinsically.

Impact on Fortran

MLIR is already being adopted into flang (compiler for fortran). As mentioned by one of the members Flang is becoming a subproject of LLVM and MLIR should be part of LLVM. Obviously there would be some redundancy and eventually flang as a part of LLVM would

be something simpler than what it is now. As MLIR is not limited to just ML applications but actually a state of the art infrastructure to improve optimisations and better capture high level ideas into IR by progressively lowering the representation, it is clearly a value addition to flang and with both of them now becoming a part of LLVM the development will definitely be more streamlined now.