15-745, Spring 2022
Project Milestone Report and Meetings

Report Due: Thursday, April 14th, 8:00am
Meetings: During regular class slot on that day

Please write up a brief (i.e. roughly one page) report of your progress so far in your project. Your report should include the following information:

Major Changes: *Have there been any major changes in the goals or implementation of your project since your project proposal? If so, please describe these changes and what brought them about.*

The goal still remains the same. What we're doing is to implement the algorithm and pass mentioned in the paper, and see if we can successfully reduce code size of real-world code(also provided in the paper). But we will slightly change how we are going to evaluate the result. We aim to test the Angha benchmark (which contains functions from real-word repositories), if somehow that does not work, we will use our self-written unit tests to verify; if it works well, we will use benchmark Mibench to apply our test on a whole program.

What You Have Accomplished So Far: *Describe exactly what your group has accomplished so far. If you have any results so far, please post them to your web page, and point us to them. (You do not need to include your results in this writeup provided that they are available on your web page.)*

So far, we managed to finish several parts of the original goal we set. We set up our development environment and did more research on related literature. We also read the code of LLVM's loop rerolling transform pass (this is different from the method that we are going to implement in that it cannot handle any form of straight code).

Meeting Your Milestone: *Did you meet the milestone which you described in your original project proposal? If not, then please explain why.*

In our project proposal, we are supposed to be implementing the final steps mentioned in the paper and would be testing with the benchmarks after the submission of the milestone report. Therefore, we did not meet our milestone, this is due to the tight schedule and heavy workload we faced in the past month and we are striving to catch up our supposed schedule.

Surprises: *Have there been any major surprises so far in your project? If they were bad surprises, how have you managed to work around them?*

We surprisingly found that the original paper open-sourced its implementation of patterns that it introduced as LLVM loop rolling passes. (He integrated in the original llvm project and rebuilt the whole project to test) However, it's unrealistic to switch to another topic at this point. So we will keep this original topic and ignore the code that the authors published.

Revised Schedule: *Describe what each member of your group is going to accomplish during your remaining time to successfully complete your project. If you are stuck on anything, please let us know. (In fact, if you are stuck, please send us an email immediately rather than waiting for us to read this report.)*

Both of us will start running experiments on the test suites mentioned in the paper and would probably need to see whether our implementations go wrong or need improvement. If we aren't able to test the benchmarks, we may turn to test it on smaller unit tests to verify the correctness of the algorithm.

We had a detailed discussion just recently to try to figure out a viable solution so that we could fulfill the original goal in the coming couple of weeks. And we've come up with a plausible roadmap towards our goal. We split the whole process into 3 phases. The first phase is alignment graph generation, which includes the collecting and grouping of seed instructions, the construction of alignment graph according to the pseudo code given in the paper, and finding out the mismatches in the alignment graphs. The second phase is mismatching checking. We try to match these mismatches with the 6 patterns and find those mismatches where loop rolling can actually be applied. Finally, for the 3rd phase, we change the original code by introducing loop rolling to those places that we find.

Resources Needed: *Do you have all of the resources (e.g., software, benchmarks, simulators, etc.) that you need to complete your project? If not, then please explain how and when you will acquire these resources.*

Although we did not have the software resources used in the project, we decided to move forward with the docker environment used in class.

The benchmarks mentioned in the paper (listed below) are all open-sourced and we can acquire them. (AghaBench and TSVC are provided in the paper's artifact), the other two, MiBench and SPEC 2017 are open-sourced.

We're still figuring out how to use the benchmark SPEC 2017, but if we fail to figure it out, we can still use only MiBench to verify the loop rolling algorithms on full programs.

Please submit your report via Gradescope. Meetings will take place via Zoom. We will use the same time slots that we used for the original project discussion meetings.