Izmir Institute of Technology
Computer Engineering Department
CENG513 Midterm Exam Spring 2024
Question 3

Student Name: Gökay Gülsoy Student No: 270201072

April 19, 2024

Question 3

LLVM projects mentioned in the talk can be listed as follows:

- clang
- clang-tools-extra
- lld

LLVM runtimes mentioned in the talk can be listed as follows:

- compiler-rt
- libcxx
- libcxxabi
- libunwind

Main difference to build LLVM projects and runtimes is that LLVM projects are built using the host compiler, whereas LLVM runtimes are built using the just-built compiler (compiler that is built when LLVM is built).

Problem for building runtimes is that runtimes contain cyclic dependencies, specifically compiler-rt runtime is required to build libunwind, libc++, libc++ABI, but compiler-rt itself also contains other runtimes which use libc++, libc++ABI, or libunwind. Proposed solution is to first build builtins as a seperate build, then build the remaining runtimes.

Ways to build LLVM faster are as follows:

1.) Using release mode make build faster than debug build because debug build produce large amount of debug info.

2.) optimizing `TABLE_GEN` with `LLVM_OPTIMIZED_TABLEGEN`, using lld as a linker with `LLVM_ENABLE_LLD` which is usually faster than other linkers. Also enabling libcxx with `LLVM_ENABLE_LIBCXX` usually speeds up build.

3.) Using LTO,PGO,BOLT builds can speed up build by 25-50%

4.) Building with less code by only building what is needed can speed up build process which can be done via setting `LLVM_TARGET_TO_BUILD` to Native which is going to create a backend only for host computer architecture. Setting `LLVM_ENABLE_PROJECTS` only to projects that will be used, and setting

`LLVM_ENABLE_RUNTIMES` only to runtimes that will be used can speed up the build.

5.) Leveraging caching with tools like ccache and distcc can speed up builds.

It would be better to watch the video to understand what specific options are existing to enable or disable certain projects and runtimes, and how the LLVM project can be configured for faster build by only building parts of LLVM which will be used.

Options to be used with cmake to configure LLVM project are clearly indicated in the talk but specifically in the part which is about building LLVM faster, it was said that LTO, PGO, BOLT builds can speed up build, but it was not explained that what are these builds and how they can basically speed up the LLVM build process.