New LLVM
Switch-per-Function Feature

May 2023
Function Attributes

- Attributes allow users to control the functionally, optimizations, code generations, and compiler decisions of the function.

- For example, the "minsize" attribute keeps the function code size as small as possible:

```c
void __attribute__((minsize)) foo0(int *p, int *q) {...}
```

- Disadvantage:
  - Forces the user to modify the source code.
Switch-per-Function

► Control optimizations at the function level from the command-line:
  ▶ No source code modify required
  ▶ Easy to configure
  ▶ Usage example:

```
-mllvm -cxd-switch-per-function="foo0=-fmin-size"
```
CEVA Implementation Steps

1. Add dedicated command-line option
2. Parse the command-line string
3. Map switches to attributes
4. Add attributes to relevant functions
CEVA Implementation Steps

1. Add a dedicated command-line option in `clang\lib\CodeGen\TargetInfo.cpp`:

   ```cpp
   static llvm::cl::opt<AttrPerFunction> SetFuncsAttrsTable("cxd-switch-per-function", llvm::cl::Hidden,
   llvm::cl::desc("Set attrs per function"));
   ```

2. Override the `setTargetAttributes` function from the `TargetCodeGenInfo` class to parse the string following the `-cxd-switch-per-function="..."` options.
CEVA Implementation Steps

3. Map each option from the parsed string to the matching function and the relevant attribute.
   - The main data structure is a map from `switch syntax` to `attribute syntax` and `bool value`, representing whether the switch should have a value.

```
SwitchToAttrMap = {
    "-fmin-size", std::make_pair("minsize", false)
};
```
CEVA Implementation Steps

4. Add each attribute to its relevant function:

```cpp
auto It = SwitchToAttrMap.find(SwitchName);
std::string AttrName = It->second.first;
if (It->second.second) {
    Fn->addFnAttr(AttrName, SwitchValue);
} else {
    Fn->addFnAttr(AttrName);
}
```
Adding New Attributes

- Some switches do not have corresponding attributes.
- We added corresponding attributes to these selected switches.
- The Switch-per-Function feature is now also available for these switches.
- We have found this extremely essential for optimizing customer code.
Suggestion for Future Support

- Add a mechanism to define, in one place, an option that will become both a global command-line option as well as a function attribute that will apply this option to a specific function:
  - Auto-generate the mapping between a switch and an attribute
  - Easier to maintain
Summary

1. **Disadvantages** of function attributes

2. **Solution**: Switch-per-Function feature

3. Switch-per-Function feature **implementation**

4. **Expand range** of switches per function

5. **Suggestion** for future support