Minimum Size Function Coverage

Ellis Hoag
Kyungwoo Lee

-preinline-threshold=5

- Extra inlining pass added before PGOGen
  - https://reviews.llvm.org/D21405
  - Default inline threshold = 75
- Threshold should match optimization mode

```
// llvm/include/llvm/Analysis/InlineCost.h

/// Use when minsize (-Oz) is specified.
const int OptMinSizeThreshold = 5;

/// Use when -O3 is specified.
const int OptAggressiveThreshold = 250;
```

`-enable-post-pgo-loop-rotation=false`

- Loop rotation transformation added after PGOGen
  - https://reviews.llvm.org/D34085
  - Can increase code size
  - Not useful for function entry coverage

-pgo-function-size-threshold

- Do not instrument small functions
  - #(LLVM-IR Instructions) < Threshold
- Imprecise
  - Good enough to identify very small functions

`noprofile` & `skipprofile`

- `Attribute::NoProfile`
  - ✅ Prevents instrumentation
  - ❌ Prevents **inlining** if callee/caller disagree on attribute
    - Needed to guarantee safety
    - Disables most inlining

- `Attribute::SkipProfile`
  - ✅ Prevents instrumentation
  - ✅ No restrictions on inlining
    - No codesize/performance surprises

-fprofile-list

- Special case list to specify functions and files to instrument

```
[llvm]

# Attribute::SkipProfile
# Block C++ standard library functions
function:_ZSt*=skip

# Attribute::NoProfile
source:lib/unsafe/*.cc=forbid

# Otherwise we allow profiling.
default:allow
```

-fprofile-function-groups

- -fprofile-function-groups=<N>
  - Partition functions into N groups
- -fprofile-selected-function-group=<i>
  - Only instrument functions in group i
- Collect profiles independently from each group
- Merge profile offline

```
$ clang++ -Oz -fprofile-generate=g0/ -fprofile-function-groups=3 -fprofile-selected-function-group=0 code.cc -o code.0
$ clang++ -Oz -fprofile-generate=g1/ -fprofile-function-groups=3 -fprofile-selected-function-group=1 code.cc -o code.1
$ clang++ -Oz -fprofile-generate=g2/ -fprofile-function-groups=3 -fprofile-selected-function-group=2 code.cc -o code.2

$ llvm-profdata merge -output=code.profdata g*/*.profraw
```

Clang-16 Size Overhead