gcov 和 clang 的实现

MaskRay

https://maskray.me

1 Code coverage

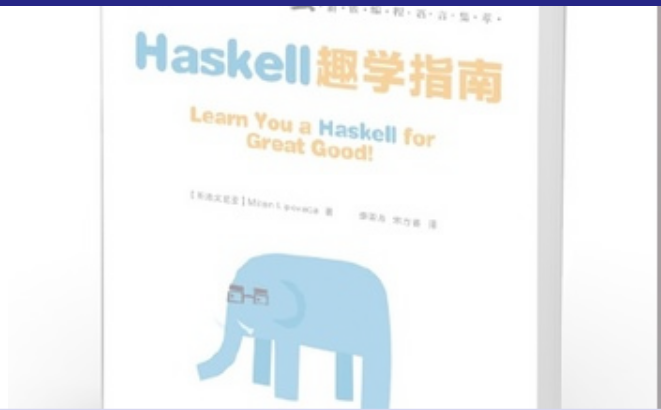
2 gcov

3 gcov in Clang

4 Linux kernel

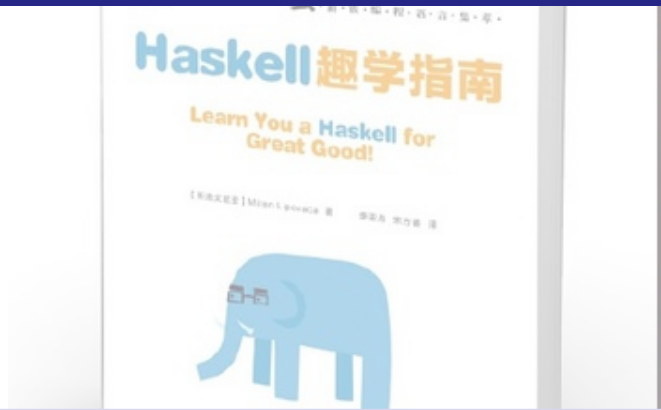
5 Future work

6 References



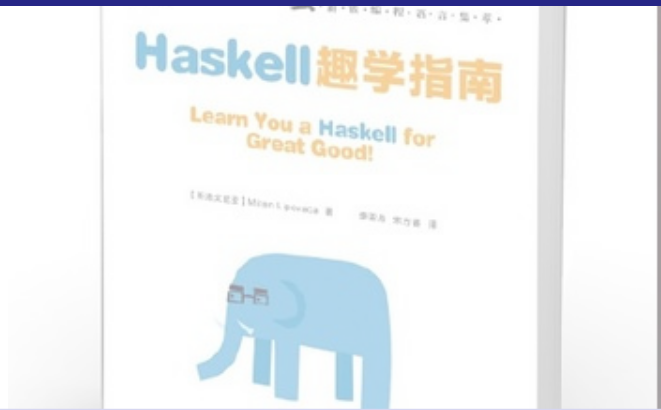
MaskRay

- LLVM contributor



MaskRay

- LLVM contributor
- ccls owner (C++ language server)



MaskRay

- LLVM contributor
- ccls owner (C++ language server)
- 退休的算法竞赛 + 超算 +CTF 选手

MaskRay

- 退休的算法竞赛 + 超算 +CTF 选手

Motivation

- Found a Bug

Motivation

- Found a Bug
- – Why is it not covered?

Motivation

- Found a Bug
- – Why is it not covered?
- Find dead code

Code coverage tools

- `gcc --coverage` + gcov
- Clang coverage mapping `clang -fprofile-instr-generate -fcoverage-mapping` + llvm-cov
- `clang -fsanitize-coverage=trace-pc-guard,{edge,bb,func} -fsanitize={address,memory,thread,...}` + sancov

gcov

- `gcc -fprofile-arcs -ftest-coverage a.c b.c # --coverage`
- compile-time `-fprofile-arcs` ⇒ a.gcno b.gcno (notes file)
- `./a.out` ⇒ a.gcda b.gcda (count data file)
- `gcov a.c # or a.` ⇒ a.c.gcov
- `gcov b.gcno # or b.` ⇒ b.c.gcov
- `gcov -h`

lcov

- `https://github.com/linux-test-project/lcov`, gcov's graphical front-end
- `lcov --gcov-tool gcov -c -d . -o a.info`
- `genhtml a.info -o html`

```
% lcov --gcov-tool gcov-9 -c -d . -o a.info
Capturing coverage data from .
Found gcov version: 9.3.0
Using intermediate gcov format
Scanning . for .gcda files ...
Found 2 data files in .
Processing b.gcda
Processing a.gcda
Finished .info-file creation
% genhtml a.info -o html
Reading data file a.info
Found 2 entries.
Found common filename prefix "/tmp"
Writing .css and .png files.
Generating output.
Processing file c/a.c
Processing file c/b.c
Writing directory view page.
Overall coverage rate:
  lines......: 100.0% (4 of 4 lines)
  functions..: 100.0% (2 of 2 functions)
```

Compatibility

- .gcno/.gcda format changed several times: GCC 3.4, 4.7, 4.8, 8, 9

```
% gcc-9 --coverage a.c; ./a.out; gcov-10 a.
a.gcno:version 'A93*', prefer 'B00e'
a.gcda:version 'A93*', prefer version 'B00e'
File 'a.c'
Lines executed:100.00% of 2
Creating 'a.c.gcov'

% gcov-8 a.
a.gcno:version 'A93*', prefer 'A84*'
a.gcno:no functions found
a.gcda:version 'A93*', prefer version 'A84*'
```

gcov in Clang

- `clang --coverage # -fprofile-arcs -ftest-coverage`
- `clang --coverage -Xclang -coverage-version='408*'` gcov 4.8 7 compatible
- instrumenter (write .gcno, generate calls into runtime) ⇔ gcc/coverage.c
- runtime (write .gcda) ⇔ libgcov.a (libgcc/libgcov-driver.c & friends)
- llvm-cov gcov ⇔ gcov

```
2020-05-10 [gcov] Fix .gcda decoding and support GCC 8, 9 and 10
2020-05-10 [gcov] Don't skip leading zeros when reading a string
2020-05-10 [gcov] Temporarily unsupport host-byteorder-big-endian
2020-05-10 [compiler-rt][test] Add feature host-byteorder-big-endian
2020-05-10 [gcov] Temporarily unsupport host-byteorder-big-endian
2020-05-10 [gcov] Delete CC1 option -coverage-no-function-names-in-data
2020-05-10 [gcov] Default coverage version to '407*' and delete CC1 option -coverage-cf
2020-05-10 [gcov] Implement --stdout -t
2020-05-11 [gcov] Emit GCOV_TAG_OBJECT_SUMMARY/GCOV_TAG_PROGRAM_SUMMARY correctly and f
2020-05-11 Revert part of D49132 "[gcov] Fix gcov profiling on big-endian machines"
2020-05-11 [gcov] Fix big-endian problems
2020-05-12 [gcov] Default coverage version to '408*' and delete CC1 option -coverage-ex
2020-05-12 [gcov][test] Fix clang test
2020-06-03 [gcov] Improve .gcno compatibility with gcov and use DataExtractor
2020-06-03 [gcov] Delete XFAIL: host-byteorder-big-endian
2020-06-03 [gcov] Make `Creating 'filename'` compatible with gcov
2020-06-03 [gcov] Don't error 'unexpected end of memory buffe'
2020-06-06 [gcov] Support big-endian .gcno and simplify version handling in .gcda
2020-06-06 [gcov] Delete `XFAIL: host-byteorder-big-endian` for test/
Transforms/GCOVProfiling/{exit-block.ll,function-numbering.ll}
2020-06-06 [gcov] Delete unneeded code
2020-06-06 [gcov] Improve tests and lower the minimum supported version to gcov 3.4
2020-06-07 [llvm-cov] Fix gcov version detection on big-endian
2020-06-07 [gcov][test] Delete UNSUPPORTED: host-byteorder-big-endian from test/
profile tests
2020-06-07 [gcov] Fix instrprof-gcov-__gcov_flush-terminate.test
2020-06-07 [gcov] Support .gcno/.gcda in gcov 8, 9 or 10 compatible formats
2020-06-09 [gcov][test] Add mkdir -p %t && cd %t
2020-06-16 [gcov] Refactor llvm-cov gcov and add SourceInfo
2020-06-16 [gcov] Add -i --intermediate-format
2020-06-16 [llvm-cov gcov] Don't suppress .gcov output if .gcda is corrupted
2020-06-17 [llvm-cov gcov] Support clang<11 fake 4.2 format
2020-07-01 [gcov] Move llvm_writeout_files from atexit to a static destructor
```

```
# Position in the legacy pass manager pipeline
% clang --coverage -mllvm -debug-pass=Arguments -c a.c
Pass Arguments:  -tti -targetlibinfo -ee-instrument
Pass Arguments:  -tti -targetlibinfo -assumption-cache-tracker -profile-summary-info -insert-
gcov-profiling -strip -forceattrs -basiccg -always-inline
Pass Arguments:  -tti -targetlibinfo -targetpassconfig ... -livedebugvalues -x86-seses -cfi-
instr-inserter -x86-lvi-ret -lazy-machine-block-freq -machine-opt-remark-emitter
# Close to -finstrument-functions, -fprofile-instr-generate
```

在 new pass manager 中的位置差不多。问题：instrument 部分位置太靠前了，而生成的函数必须提前来利用后续的优化

Instrumenter

- 添加一个 static constructor `__llvm_gcov_init`，呼叫 runtime `llvm_gcov_init`
- 每个函数：`@__llvm_gcov_ctr = internal global [$n x i64] zeroinitializer`, `$n` 为边数
- basic block 转移时插入指令修改 `__llvm_gcov_ctr` 元素
- `__llvm_gcov_writeout`：输出所有 `@__llvm_gcov_ctr`

`llvm_gcov_init`

- 定义在 `libclang_rt.profile-x86_64.a(GCDAProfiling.c.o)`
- 接收 writeout, flush, reset 三个 callbacks
- writeout: 写 `.gcda`
- reset: 清零 counters
- `__gcov_flush`=writeout+reset (无用，GCC 11 被移除)
- 设置 `atexit` hook，在程序退出时呼叫 writeout

llvm-cov gcov

- gcov 3.4 10 + clang's fake gcov 4.2 format
- `-i, --intermediate-format`
- `-t, --stdout`

Linux kernel 的 .gcda runtime

- kernel/gcov/gcc_4_7.c
- kernel/gcov/clang.c

Future work

- 减少 counters 数目：后移 pass
- 实现目前忽略的字段
- gcov 8 模仿 coverage mapping，可以显示不同 template instantiations 的行计数
- llvm-cov gcov 准确支持 GCC>=9 行计数

References

- https://gcc.gnu.org/onlinedocs/gcc/Gcov.html