

Coverage-directed codebase reduction for the procedural generation of lit tests

Freya Fewtrell
EuroLLVM 2026

Motivation

Current workflow

Large real-world build

Crash/miscompile found

Reduce to create test case

File bug report upstream

Desired workflow

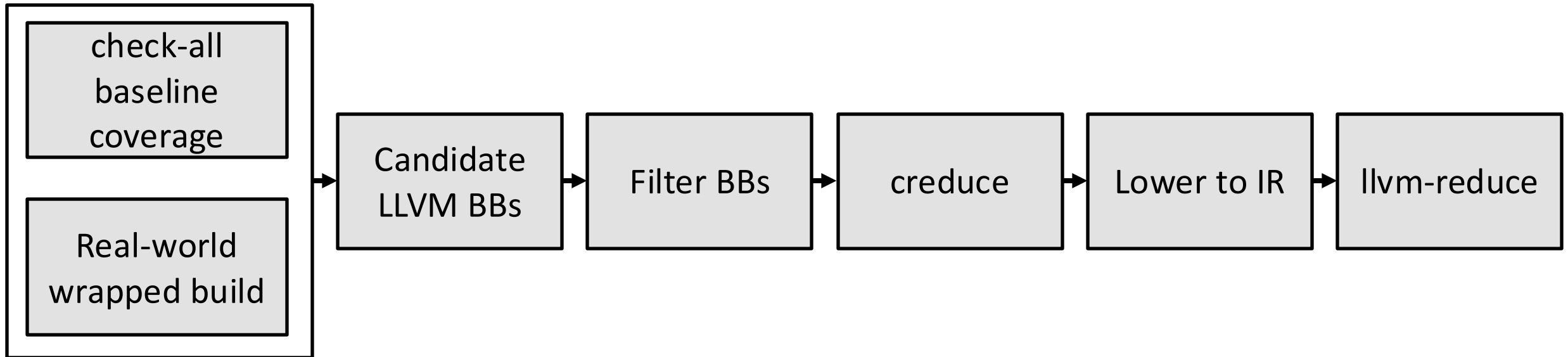
Large real-world build

New LLVM coverage found

Reduce to candidate test

Upstream before any failures

Alecto pipeline



Interestingness test

```
#!/bin/bash -e
```

```
TARGET="0x0aa1d14d5e6a4f6c 17
```

```
DAGCombiner.cpp:_ZN12_GLOBAL__N_111DAGCombiner28reduceBuildVecTruncToBitCastEPN41lvm6SDNodeE"
```

```
LLVM_PROFILE_FILE=prof.profraw clang++ -c test.cpp -O3
```

```
llvm-profdata merge prof.profraw -o prof.profdata
```

```
llvm-alookup prof.profdata $TARGET
```

Where automation gets hard

Automated well

Coverage delta detection

Testcase reduction

Convert to lit test

Still manual

Identifying genuinely useful tests

Open question