FERMILAB-SLIDES-19-035-CD

Static Analyzer Non-Comprehensive Overview

Dr Christopher Jones

HOW 2019

21 March 2019

This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics

Purpose of Talk

- Provide an overview of some of the code static analysis done by experiments

- Not a comprehensive list
  - I only contacted people I knew
  - Any mistakes in the information presented are mine
  - I list all CMS ones though other experiments may have similar checkers

- Meant to start a discussion in the meeting

Compiler

- Experiments known to use: all

- Warnings from the compiler are a form of static analysis
- Many experiments use multiple compilers or versions of a compiler
  - clang and gcc seem to be the most popular

Coverity

- Experiments known to use: ATLAS, CMS, LHCb

- Commercial package
  - https://www.synopsys.com/software-integrity/security-testing/static-analysis-sast.html
  - CERN has a license

- Provides a wide selection of sanity and correctness checking for C++
  - improper memory handle
  - many kinds of resource leaks
    - failing to release file handles
  - threading problems
    - deadlocks
    - improper locking

- Has had problems keeping up with the C++ standard
  - All known experiments have temporarily stopped using it because of this

Codacy

- Experiments Known to Use: ALICE

- Commercial Tool
  - https://www.codacy.com
  - https://github.com/marketplace/codacy

- Provides tools for automating code reviews
  - Uses a plugin system to run different tools for multiple languages
    - cppcheck
    - flawfinder
    - Pylint

- Easy integration with GitHub
- Nice reporting tools

cppcheck

- Experiments known to use: ALICE, ATLAS

- Open Source
  - http://cppcheck.sourceforge.net

- Reports bug in C/C++ with an emphasis on undefined behavior
  - dead pointers
  - integer overflows
  - invalid use of STL

clang-tidy

- Experiments Known to Use: ALICE, CMS

- Open source
  - https://clang.llvm.org/extra/clang-tidy/
  - stand alone executable

- Can diagnose and in some cases fix typical programming errors
  - add `override` keyword
  - change comparison of `std::string` to “” to call to empty()
- Very customizable via configuration
- Can be extended
  - Examples from ALICE
    - enforce member data naming convention
    - catch cases where `sizeof` should be used

clang Static Analyzer

- Experiment known to use: CMS

- Open source
  - Plugins loaded by the clang compiler

- Uses exhaustive program-flow to try to find problems
  - returning null reference
  - dead assignment
  - memory leaks
- CMS extensions
  - using namespace in headers
  - lots of thread safety checks
    - global variables
    - const member functions returning non-const pointers to member data
  - Use thread-safety report in conjunction with a graph of what functions call other functions to find all Framework modules associated with ‘global’ variables

gcc plugin

- Experiment known to use: ATLAS

- Open source
  - plugins loaded by the gcc compiler

- ATLAS uses
  - enforcing naming conventions
  - flagging thread-unsafe constructs
    - mark code as being required to be thread safe using C++ annotations
    - marked code can only call other marked code

Include What You Use

- Experiment known to use: CMS

- Open source
  - https://github.com/include-what-you-use/include-what-you-use
  - based on clang

- Can identify and fix incorrect includes
  - unneeded headers
  - missing direct includes for cases where functions/classes are indirectly included

gcc libCheck

- Experiment known to use: CMS

- Open source
  - gcc using -as-needed flag

- makes linker say which linked libraries were unnecessary

CMS Homegrown

- Package dependency checker
  - packages are the smallest unit CMS uses to compile
  - attempt to enforce allowed dependencies between groups of packages
    - e.g. Reconstruction code should not dependent on simulation

- Checks for ROOT dictionaries
  - find duplicate ROOT dictionaries across packages
  - find dictionaries defined in a package not containing the C++ class
  - catch class changes without corresponding ROOT version number change