GO2 TEAM MEETING #3

JAN 23, 2020

Weekly Progress

1. Run our tool on several Kubernetes packages, and produced data race results.
2. Revise the tool to incorporate channel Ops and Sync Order.
3. Modify the algorithm for Program Order verification.

Data Race Detection Using Static Analysis

Information Retrieval

- The packages are loaded in `doAnalysis` where an SSA program is constructed.

- The accesses of each function is recorded by obtaining ***basic blocks*** of the function
- Individually examine every ***instruction*** within a basic block
- Sync blocks are extracted from basic blocks at every ***synchronization operation***.

Analysis

- Conduct pointer analysis using the built-in pointer package which implements Andersen's points-to analysis
    - translate statements "p = q" to constraints of the form "q's points-to set is a subset of p's points-to set"
    - flow and context insensitive
- Obtain callgraph

- Take every pair of functions, compare every pair of accesses, and obtain conflicting pairs
  - can run in parallel
  - at least one write instruction
  - at most one atomic instruction
  - access same memory location
  - can't establish happens-before

TODO:

- Improve PO and SO verification
    - PO - manipulate dominator tree of CFG
    - SO - consider channel "peers"
- Consider other synchronization primitives - i.e. WaitGroup, etc.

Start at actual entry point of k8s, conduct profiling

prepare tutorial on K8s.