2019-01-29 - golang-tools session - meeting notes

Previous session notes

2018-12-11 - golang-tools session - meeting notes

Recording

https://www.youtube.com/watch?v=mKAj_6ZbvfM

Attendees

- Hana Kim
- Michael Matloob
- Jay Conrod
- Ian Cottrell
- Bryan Mills
- David Chase
- Daniel Marti
- Roger Peppe
- Marwan Sulaiman
- Chris Wendt
- Peter Waller

Notes

- mvdan: using go/packages more often, filed bugs, porting tools to go/analysis, but need callgraph.
- matloob: changing the name= query in go/packages. No one is using it except goimports, and it's not working well, so we want to hide it a little bit.
- iancottrell: missing tool to analyze init phase of applications.
    - bcmills: We can get part way there with Linux perf tool.
    - iancottrell: Was thinking more of static analysis, lint to point out something that will take a long time.
    - hakim: regular profiling / tracing is usually initialized in main, so it's a little difficult to do.
- iancottrell: LSP tool now called gopls. Almost ready for use. Rebecca will send an email to golang-tools this week when it's ready.
    - iancottrell: growing some verbs to behave like guru. The intention is to deprecate guru completely and switch to gopls.
    - Peter Waller: is it straightforward to try with VSCode?
    - iancottrell: In settings, there is an override for tools. Tell VSCode to use it as the language server. Rebecca's email will contain instructions.
    - rogpeppe: Is there a good guide for getting started from scratch?
    - iancottrell: Not any really good guides out there yet. Expect at some point every editor will need some support for it though. Expectation is that the editor runs it as a subprocess and communicates over stdin/stdout. There's a lot of complexity in running a daemon, especially in multi-user situations, so we've stayed away from that so far.
    - rogpeppe: Does it do goimports?

    - iancottrell: It's a save hook for now. Rebecca will explain. For now, we've provided it as a source code action. Separated from formatting.
    - mvdan: If someone has LSP installed and an editor that supports it, will the editor still call go list directly?
    - iancottrell: LSP has no means to run builds or tests. So VSCode commands for running test will need to talk directly to the go command. Editor may use go list or go env, but need to be aware of other build systems.
    - mvdan: A while ago, we spoke about a lower level API than go/packages. Will that still be published?
    - matloob: We're thinking of a more fine-grained way to choose what information go/packages does. There's a bug. If enough people think it's reasonable, we'll do that.
    - marwan-at-work: Is gopls + go/packages performance still tied to go list?
    - matloob: gopls will call into go/packages once initially, then will operate incrementally on its own after that.
    - iancottrell: It knows about changes in the editor, so it can invalidate old results. Next time you need a new result, it will query go/packages again. So it's faster for repeated queries, but slow for new stuff. We'll try to run go/packages earlier to hide the latency. There's a lot of room for improvement though.
    - rogpeppe: How much memory does it take?
    - iancottrell: Probably a ridiculous amount, but not anymore than gocode was already using. Something we'll have to keep an eye on. There is a tradeoff between responsiveness and memory use.
    - mvdan: Is there a plan to make go/types, go/ast etc use less memory?
    - iancottrell: Would like to see that done, but not that high up the priorities. Some big questions about whether we should go through a massive API change unifying compiler and stdlib types.
- bcmills: We're planning to turn on modules by default in 1.13. It would be great if all the tools can be module enabled for 1.13.
    - mvdan: Shouldn't the documentation be improved?
    - bcmills: It's one of the larger clusters of issues to be done. We have a lot of GOPATH-mode documentation to be updated, go modules docs have grown organically, we need to get the commands working properly so we don't have pages for workarounds for known issues.
- rogpeppe: How go/parser reacts to malformed source code? godef breaks if there's a syntax somewhere.
    - iancottrell: Rebecca has been working on this for the completion engine. It's more resilient than it used to be. Every time you find an issue like this, talk to her.